

Universidad de las Ciencias Informáticas
Facultad 2



Título:

***Diseño e Implementación del módulo Evaluación
Criminológica perteneciente al SIDEPA.***

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autora: Adys Aragón Rodríguez.

Tutores: Ing. Juan Carlos Gómez.

Ing. Maikel David Pérez.

La Habana, 2012.



“La ciencia se compone de errores, que a su vez, son los pasos hacia la verdad.”

Julio Verne

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año_____.

Adys Aragón Rodríguez.

Ing. Juan Carlos Gómez.

Ing. Maikel David Pérez.

Dedicatoria

A mi madre por su amor incondicional y su apoyo constante.

A mi padre por su ejemplo y por su inmenso cariño.

A mis abuelos por la educación recibida.

A mi novio por haber estado siempre ahí para mí.

A mi familia en general por todos los momentos que hemos pasado juntos.

Agradecimientos

A la Revolución y a la Universidad de las Ciencias Informáticas por la oportunidad de estudiar y de cumplir una meta tan grande.

A mi madre Ada por ser esa luz que me guía por buenos caminos, por su amor infinito, por sostenerme con su mano fuerte en los momentos difíciles y de debilidad, por su comprensión, por ser una mujer maravillosa a la que admiro por encima de todas las cosas y por alentarme a seguir avanzando a pesar de los obstáculos que la vida se empeña en ponernos en el camino.

A mi padre Miguel por su amor, por su ejemplo de lucha y sacrificio, por su ayuda en todo momento, por tener siempre un consejo oportuno y unas palabras de ánimo para darme.

A mis abuelos Laura y Edelso por todo el cariño que he recibido de ellos, por haberme dado gran parte de la educación que tengo y que me hace ser la persona que soy, por haber estado siempre cerca y por ser mis segundos padres.

A mis tíos Maité y Edelsito por haberme cuidado tanto y por haber cumplido mis caprichos por mucho tiempo, por ser los mejores del mundo.

A mi novio José Alberto por su amor, su compañía y por apoyarme todo el tiempo.

A mi hermana por haber compartido conmigo sus travesuras y espero poder ser un ejemplo a seguir por ella en el futuro.

A mi familia en general por ser todos maravillosos y por estar siempre dispuestos a enfrentar todos los momentos unidos.

A mi otra familia, Aracelia, Amarelis, Pepe, María Antonia, Aracelita, a todos gracias por haberme acogido en su seno durante todos estos años y por haber cuidado de mí en los momentos que lo necesitaba.

A mis tutores Maikel y Juan Carlos, por haber sacado el tiempo a pesar de las muchas tareas que siempre tuvieron, por ayudarme cuando me parecía que no podía terminar todas las cosas que tenía que hacer y por darme ánimo para seguir.

A los demás integrantes del SIDEPE especialmente a Yanet, Reinier, Yaidel, Daniel, a todos los que de una forma u otra contribuyeron con la realización de este trabajo, muchas gracias.

A los amigos que me han acompañado durante toda mi vida y a los que han compartido conmigo estos 5 años de estudio y sacrificio. A Isel mi amiga desde que puse un pie en la UCI, la pobre que ha tenido que soportarme por mucho tiempo. A Víctor por ayudarme y por ser una persona que me ha hecho reír en los momentos más tristes. A Irina, Meralys y Maidelis por haberme apoyado en el momento que más lo necesitaba, cuando estaba lejos de mi familia.

A mis compañeros de aula durante estos 5 años y a los integrantes de la casa de estudio del aula de CDI Reinier, Rayko, Yaidel, Ricardo, Ale, e Irina.

A mis compañeras de apartamento Leibys, Lisandra, Yordanka, en fin a todas por el tiempo que hemos compartido.

A todos los profesores que han contribuido a mi formación como profesional.

Resumen

Actualmente el Sistema Penitenciario cubano utiliza el SACORE (Sistema Automatizado para el Control del Recluso) como solución informática para la gestión de la información de los internos. Aunque este sistema automatiza algunos de los procesos que se realizan en los Centros Penitenciarios no sucede así con el proceso de evaluación criminológica, a través del cual se evalúa a los internos durante su estancia en el Área de Ingreso. El objetivo final de este proceso es elaborar un plan de tratamiento individualizado que permita la rehabilitación del interno dirigido a su reinserción en la sociedad.

La evaluación criminológica implica el almacenamiento y manipulación de una gran cantidad de documentos, lo que trae consigo la pérdida de caracterizaciones y de entrevistas realizadas a los internos, y deficiencias en la correcta realización del proceso. Todo esto influye negativamente en la realización del trabajo diferenciado que se debe hacer con los internos en los Centros Penitenciarios.

El presente trabajo consiste en el diseño e implementación del módulo Evaluación Criminológica perteneciente al Sistema Informativo de la Dirección de Establecimientos Penitenciarios (SIDEPE), con el objetivo de facilitar la realización del proceso de evaluación criminológica. Durante el desarrollo de este trabajo se cumplió con los objetivos y tareas definidas a partir del análisis de los requerimientos de software establecidos de acuerdo con el cliente, utilizando las herramientas y tecnologías seleccionadas.

Luego de finalizado el trabajo se obtuvo como resultado una aplicación que dota al SIDEPE de un mecanismo para facilitar la gestión del proceso de evaluación criminológica en los Centros Penitenciarios cubanos.

Tabla de contenido

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.2. Términos Clave	5
1.3. Estado del arte	6
1.3.1. Lock&Track Corrections Information System	6
1.3.2. Elite Jails	7
1.3.3. Establecimiento Penitenciario Virtual en Red (España)	8
1.3.4. Sistema de Gestión Penitenciaria (SIGEP)	9
1.3.5. Sistema Automatizado para el Control del Recluso (SACORE).....	11
1.3.6. Resultados del estudio del arte.	11
1.4. Metodología, tecnologías y herramientas utilizadas.....	12
1.4.1. Metodología de desarrollo.....	12
1.4.2. Herramientas de Modelado	14
1.4.3. Sistema Gestor de Base de Datos Oracle 11g	15
1.4.4. Herramientas de desarrollo	15
1.4.5. Tecnologías y lenguajes.....	17
1.5. El proceso de Evaluación Criminológica.	21
1.6. Conclusiones Parciales	22
CAPÍTULO II: DISEÑO DEL SISTEMA	24
2.1. Introducción	24
2.2. Descripción del módulo Evaluación Criminológica	24
2.3. Propósito del Diseño	29
2.4. Modelo de Datos.	29
2.5. Diagrama de Clases del Diseño	30
2.6. Patrones de diseño utilizados.....	32
2.6.1. Modelo Vista Controlador (MVC)	32
2.6.2. Singleton	32

2.6.3.	Patrón Experto	33
2.6.4.	Patrón Controlador	33
2.7.	Diagramas de Interacción. Diagrama de Secuencia	34
2.8.	Flujo Web (Web Flow)	36
	Conclusiones Parciales	38
CAPÍTULO III:	IMPLEMENTACIÓN Y PRUEBA	39
3.1.	Introducción	39
3.2.	Implementación	39
3.2.1.	Implementación de las clases del Dominio	39
3.2.2.	Implementación de las clases Controladoras	41
3.2.3.	Implementación de las clases Servicios	43
3.2.4.	Creación de Entidades	44
3.2.5.	Operaciones sobre el Modelo de Datos	45
3.2.6.	Internacionalización	45
3.2.7.	Diagrama de componentes	46
3.2.8.	Diagrama de despliegue	49
3.3.	Pruebas	51
3.3.1.	Método de pruebas: Pruebas de Caja Negra	51
3.3.2.	Técnica de pruebas: Diseño de casos de prueba	51
3.3.3.	Nivel de Pruebas	52
3.3.4.	Diseño de casos de prueba para el CU Gestionar Información del Área Laboral	53
3.4.	Resultados	58
3.5.	Conclusiones Parciales	60
Conclusiones	62
Recomendaciones	63
Referencias Bibliográficas	64
Anexos	66
Anexo 1:	Modelo de Datos	66
Anexo 2:	Diagramas del diseño	67
Glosario de Términos	69

INTRODUCCIÓN

Como parte de la continua evolución del Sistema Penitenciario cubano y a la par con el proceso de informatización de la sociedad, en 1989 comienza la automatización de los Centros Penitenciarios con un sistema de Control Penal que permitía la gestión de los principales datos del recluso y algunos aspectos del trabajo de la especialidad. A raíz del cumplimiento de la orden 43/99 del Viceministro Primero del Interior se diseña, programa e instala el Sistema Automatizado para el Control del Recluso (SACORE) para lo que se utilizaron las tecnologías existentes en esos momentos en el MININT, la tarea concluye en el 2002 y se comienza con la explotación del sistema en enero del 2003. [18]

En sus años de explotación el SACORE ha sido perfeccionado, a partir de los requerimientos y solicitudes de los usuarios, conteniendo como módulos principales Control Penal, Tratamiento Educativo y Orden Interior. Además se desarrollaron otros dos sistemas automatizados que complementan la información del SACORE: el Sistema Automatizado de Incidencias de la Dirección de Establecimientos Penitenciarios (SAIDEP) y el Sistema Administración de Capacidades de la Dirección de Establecimientos Penitenciarios (SACDEP). [19]

A partir del Plan 20 x 50 para la modernización tecnológica del MININT y de sus establecimientos penitenciarios, y a la par del desarrollo en materia de informática y comunicaciones que se ha alcanzado en el país, la dirección del MININT decide crear, en conjunto con la Universidad de las Ciencias Informáticas, el proyecto Prisiones Cuba; el cual tiene como objetivo implementar el Sistema Informativo de la DEP¹ (SIDEPA). Esta solución informática debe integrar las funciones de los 3 sistemas existentes actualmente en la DEP, así como adicionar nuevas funcionalidades con el fin de automatizar un conjunto de procesos que no se encuentran automatizados en los sistemas que se encuentran en explotación.

Los procesos que se desarrollan en el Área de Ingreso no están automatizados en su totalidad por los sistemas existentes. El Área de Ingreso es un lugar donde se recluye a los internos luego de su ingreso a los Centros Penitenciarios por un período que no excede los 30 días. Durante este tiempo se realiza un proceso de evaluación y diagnóstico de la personalidad y, de otros factores que se deben tomar en

¹ Dirección de Establecimientos Penitenciarios

consideración para lograr un correcto desarrollo del proceso educativo individual con cada interno. A este proceso se le denomina evaluación criminológica. Los antecedentes de este proceso en el Sistema Penitenciario cubano se remontan a 1964 [4], con la aparición de los Centros de Evaluación de Menores. En estos centros a través de la psiquiatría y la psicología se evaluaba la conducta de los menores, mediante la utilización de métodos y técnicas para evaluar la conducta criminal. Esta práctica se extendió posteriormente a la población criminal adulta y a los demás Centros Penitenciarios. [4]

Actualmente en el Sistema Penitenciario se concibe la evaluación criminológica como un diagnóstico criminológico en el que a través de sucesivas aproximaciones se obtiene un conocimiento del interno, lo que significa conocer sus características generales, motivaciones, historia personal, hechos delictivos cometidos y las circunstancias en que ocurrieron, debilidades o vulnerabilidades y fortalezas.[24] El objetivo final de este proceso es elaborar un plan de tratamiento individualizado que permita la rehabilitación del interno dirigido a su reinserción en la sociedad.

La evaluación criminológica implica el almacenamiento y manipulación de una gran cantidad de documentos, lo que trae consigo la pérdida de caracterizaciones y de entrevistas realizadas a los internos, y deficiencias en la correcta realización del proceso. Todo esto influye negativamente en la realización del trabajo diferenciado que se debe hacer con los internos en los Centros Penitenciarios. Además los sistemas informáticos con los que cuenta la DEP no informatizan el proceso de evaluación criminológica.

Dada la problemática anterior se plantea como **problema de la investigación**: ¿Cómo facilitar la gestión del proceso de evaluación criminológica durante el ingreso de los internos a los Centros Penitenciarios?

Para dar solución al problema se trazó como **objetivo general**:

Desarrollar el módulo Evaluación Criminológica perteneciente al SIDE P.

Para dar cumplimiento al objetivo general se definieron los siguientes **objetivos específicos**:

- Elaborar la fundamentación teórica de la investigación.
- Diseñar el módulo Evaluación Criminológica perteneciente al SIDE P.
- Implementar el módulo Evaluación Criminológica perteneciente al SIDE P.

- Realizar pruebas de calidad al módulo Evaluación Criminológica del SIDEPE.

Definiendo como **objeto de estudio** el proceso de gestión penitenciaria en los Centros Penitenciarios y como **campo de acción**: el proceso de evaluación criminológica en los Centros Penitenciarios.

Los objetivos específicos se dividieron en las siguientes **tareas de investigación**:

- Análisis de soluciones informáticas existentes para Sistemas Penitenciarios.
- Descripción de las tecnologías y herramientas a utilizar para el diseño e implementación del módulo Evaluación Criminológica.
- Diseño de los diagramas de clases del módulo Evaluación Criminológica.
- Diseño de los diagramas de interacción del módulo Evaluación Criminológica.
- Diseño de la Base de Datos del módulo Evaluación Criminológica.
- Elaboración de los diagramas para representar el funcionamiento del flujo web.
- Implementación del módulo Evaluación Criminológica.
- Diseño de los diagramas de componentes del módulo Evaluación Criminológica.
- Diseño del diagrama de despliegue del módulo Evaluación Criminológica.
- Diseño de los casos de prueba para el módulo implementado.
- Realización de pruebas funcionales al módulo implementado aplicando los casos de prueba diseñados.

Este documento se encuentra compuesto por 3 capítulos:

Capítulo I: Fundamentación Teórica

En este capítulo se presenta un estudio de soluciones informáticas para Sistemas Penitenciarios, tanto en Cuba como a nivel mundial, centrando la atención en el proceso de evaluación criminológica. También se hace referencia a las tecnologías, las metodologías de desarrollo y las herramientas a utilizar para dar solución al problema planteado.

Capítulo II: Diseño.

Se describen los patrones de diseño utilizados, además se muestran los diagramas de clases del diseño y de secuencia, así como el diagrama del diseño de la base de datos donde se almacenan los datos pertenecientes al módulo Evaluación Criminológica.

Capítulo III: Implementación y Prueba.

Se presenta el diagrama de despliegue y el diagrama de componentes, también se implementa el sistema y se realizan las pruebas para comprobar que la aplicación cumple con los objetivos planteados.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

Durante el desarrollo de este capítulo se exponen algunas soluciones informáticas que automatizan la gestión de los procesos penitenciarios en el mundo y en Cuba. Además se definen términos clave que serán utilizados durante el desarrollo de este documento para una mejor comprensión del contenido del mismo. También se realiza una breve descripción de la metodología de desarrollo, de las tecnologías y de las herramientas que serán utilizadas para realizar el diseño y la implementación del módulo Evaluación Criminológica, justificando de esta forma su selección.

1.2. Términos Clave

Área de Evaluación y Diagnóstico: Segmento del área interior del penal que se destina para la recepción, estudio, caracterización y evaluación por las diferentes especialidades del Establecimiento Penitenciario de los ingresos o traslados de internos. Es la estructura encargada del proceso de ingreso, observación, evaluación y diagnóstico criminológico del interno, basado en el examen médico estomatológico y el estudio integral de las diferentes áreas de la personalidad, conducente a un dictamen integral de la misma y un pronóstico inicial de corrección. [15]

Proceso de Evaluación: Constituye el conjunto de medidas, acciones y exámenes dirigidos a conocer la personalidad del interno. Este proceso comprende entrevistas y evaluaciones al interno, entrevistas a familiares del interno, aplicación de pruebas o test psicológicos, orientación social, psicológica, pedagógica y jurídica a los internos mediante entrevistas de atención individual, entre otras actividades. [15]

Centro Penitenciario: Sustituye la denominación establecimiento penitenciario. Son abiertos y cerrados. Los cerrados a la vez son de Mayor y media seguridad, y los abiertos son de mínima. [15]

Interno: Denominación genérica que identifica a los acusados, sancionados o asegurados en el proceso de ejecución de la sanción de privación de libertad, medida cautelar de prisión provisional y mediada reeducativa de internamiento respectivamente. [15]

1.3. Estado del arte

Un sistema que automatice los procesos de los Centros Penitenciarios facilita enormemente el suministro y el análisis de la información, en particular para controlar el cumplimiento de la sanción de los internos y de esta forma minimizar la pérdida de tiempo del personal que realiza estas funciones.

1.3.1. Lock&Track Corrections Information System

Lock&Track Corrections Information System es una solución informática desarrollada por LockWorks LLC que permite la gestión de los procesos en las instituciones penitenciarias. Este sistema está formado por un conjunto de componentes de aplicación llamados “de escritorio”, cada uno de ellos se centra en un aspecto particular o requerimiento funcional específico en relación con el flujo de información dentro de la institución penitenciaria. Los componentes “de escritorio” constituyen el principal medio por el cual los usuarios acceden a la información relacionada con los internos. [22]

Algunas de las funcionalidades que provee este sistema son:

1. Ingreso, Admisión, Transferencia y Liberación del interno.
2. Gestión de la custodia.
3. Casos de internos y Registro de Delitos, incluidas las penas y el tiempo cumplido.
4. Generación y procesamiento de informes disciplinarios.
5. Puestos de control y seguimiento de movimientos.
6. Servicios para el interno, incluyendo médico y dental.
7. Programas de Educación para internos.
8. Cuentas bancarias de los internos.
9. Facturación interinstitucional.
10. Comisaría y Gestión de los datos almacenados.
11. Gestión de las visitas y registro de los visitantes.
12. Investigaciones internas.

13. Servicio de comida.
14. Gestión de la institución.
15. Vehículos y propiedades.
16. Análisis de la información del crimen.

Este sistema permite la integración con escáner de huellas dactilares, cámaras para la captura de imágenes y lectores de código de barras. El sistema operativo utilizado para los servidores que almacenan la aplicación es OpenVMS (Open Virtual Memory System) y el Sistema Gestor de Base de Datos es Oracle/Rdb. [22]

Lock&Track se encuentra totalmente integrado con Lock&Force Sheriff's Information System que es un sistema que está orientado a los agentes del orden de las patrullas para la gestión de incidencias y los reportes de las mismas. [22]

La combinación de estos 2 sistemas proporciona un recurso integral de gestión de la información para todos los departamentos de alguaciles, en los que se puede tener una visión completamente coordinada de las operaciones e incidentes desde el inicio del caso hasta la finalización del mismo. [22]

1.3.2. Elite Jails

Elite Jails JMS (Jail Management System) es una solución informática desarrollada por Syscon que proporciona un conjunto herramientas de software para la gestión de la información de los internos. Este sistema soporta múltiples usuarios conectados simultáneamente y ofrece un modelo de seguridad basado en roles para asegurar que cada usuario acceda y modifique solo la información para la que tiene autorización. [31]

Utiliza como Sistema Gestor de Base de Datos Oracle 10g.

Algunos de las funciones principales de Elite Jails son:

1. Ingreso: Controla la entrada o el reingreso de los internos en el sistema, mantiene un historial penal de todos los internos, incluyendo datos demográficos detallados de los mismos.
2. Viviendas: Asignación de viviendas a los internos según el criterio definido por la institución.

3. Programación y Movimientos: Mantiene un registro detallado de todos los movimientos realizados tanto internos como externos, programados y no programados.
4. Imágenes: Graba una o varias imágenes de los individuos (cara, de perfil, señales físicas) utilizando una cámara digital. La imagen captada se almacena automáticamente en un registro con los datos de los demás individuos.
5. Programas y Servicios: Se refiere a los programas específicos, tanto internos como externos, de cada sancionado; gestiona los horarios para los programas asignados, además la asistencia y el rendimiento en las actividades según los objetivos definidos y los criterios de medición de evolución.
6. Casos de la corte: Almacena las órdenes legales que autorizan la custodia del interno, y mantiene el estado de la custodia de cada interno.
7. Gestión de propiedades: Mantiene la información sobre todos los bienes que tenía el individuo al entrar a la prisión y sobre todos los recursos que le entrega la institución. También mantiene un historial de las transacciones realizadas en relación con estas propiedades.
8. Biometría: Permite el uso combinado de captura de imágenes, escáner de iris y otros identificadores biométricos.
9. Inteligencia de Negocio: Administración de los informes obtenidos a partir del almacén de datos de Syscon.
10. Incidentes y medidas disciplinarias: Se crea un caso por cada incidente reportado. Se registran detalles esenciales como: tipo de incidente, los internos o el personal involucrado en el incidente, la ubicación y las medidas adoptadas. Luego de ser guardados los detalles, se completa una evaluación del incidente. El cuestionario para el incidente y el proceso de evaluación del mismo, puede ser configurado según las especificaciones del tipo de incidente que se trata.
11. Visitas: Gestión de las visitas a los internos y del horario de las mismas, para esto es necesario una lista de las visitas autorizadas. [31]

1.3.3. Establecimiento Penitenciario Virtual en Red (España)

Establecimiento Penitenciario Virtual en Red (EPVNET) es un proyecto de software libre para la gestión informatizada de los procesos en los Centros Penitenciarios.

Tiene como precedente en España algunas aplicaciones basadas en la arquitectura cliente-servidor y bajo entorno Windows que cubren algunas áreas: nóminas de internos, empleados, la productividad de los talleres, tramitación de los expedientes penales y penitenciarios de los internos; pero con la limitante de no contemplar áreas de gestión como las que afectan al control de los movimientos de la población interna, las comunicaciones, el control de situaciones regimentales, control de accesos, etcétera.

El proyecto EPVNET es un software de gestión para la gerencia de los internos y de los empleados en cualquier centro penitenciario basado en la legislación penal española que garantiza el control de movimientos, de comunicaciones y de informes. Comenzó como un proyecto aislado en el año 2002 para dar una solución de gestión en el centro penitenciario de Valencia. Para el 2004 se convirtió en un proyecto de software libre hospedado en sourceforge.net bajo licencia GNU/GPL. Es una aplicación web desarrollada en PHP4 y PL/SQL, con PostgreSQL como gestor de base de datos. [12]

1.3.4. Sistema de Gestión Penitenciaria (SIGEP)

El Sistema de Gestión Penitenciaria (SIGEP) es una solución informática, resultado del proyecto Humanización Penitenciaria de la República Bolivariana de Venezuela, desarrollada en la Universidad de Ciencias Informáticas (UCI) a la medida de las características del Sistema Penitenciario venezolano. [25]

Las funcionalidades del software están orientadas a garantizar los siguientes objetivos:

1. Mantener identificado al privado de libertad en su tránsito por el Sistema Penitenciario.
2. Conocer la cantidad de privados de libertad, su ubicación y situación legal.
3. Conocer el comportamiento de los ingresos y egresos al Sistema Penitenciario.
4. Controlar el estricto cumplimiento de las disposiciones legales: ejecución de la pena, otorgamiento de fórmulas alternativas de cumplimiento de pena, otorgamiento de beneficios. Pronosticar libertades.

5. Conocer y valorar la incorporación al trabajo, al estudio y las actividades culturales y deportivas.
6. Registrar las personas vinculadas al interno y los tipos de influencia.
7. Controlar los planes de rehabilitación, entrevistas y diagnósticos psicológicos y sociales.
8. Controlar la aplicación de medidas disciplinarias y suspensiones de derechos.
9. Controlar las requisas y sus resultados.
10. Planificar y controlar las visitas familiares e institucionales.
11. Controlar las salidas transitorias y las presentaciones a juicio.
12. Administrar las capacidades, así como conocer las disponibilidades y afectaciones de la infraestructura física de las instalaciones.
13. Registrar de forma oportuna los hechos que ocurren en los establecimientos penitenciarios.
14. Elaborar partes informativos que sirvan de soporte al trabajo operativo de la sala situacional.
15. Controlar el estado de salud del interno. Controlar las consultas, chequeos médicos, planes de vacunación y análisis clínicos (programados y no programados). [25]

El subsistema de *Clasificación y Atención Integral* permite controlar las actividades de atención cuyo objetivo es lograr la rehabilitación y reinserción de los privados de libertad. Además se gestiona información de interés para la clasificación y las evaluaciones a las que son sometidos los individuos. Uno de los módulos definidos dentro de este subsistema es el de Clasificación:

- **Clasificación:** Finalizado el período de observación, el individuo es diagnosticado y un equipo multidisciplinario lo clasifica en alguno de los regímenes de seguridad penal (máxima, media o mínima), para ofrecerle un atención individualizada. El módulo permitirá registrar la clasificación otorgada a cada individuo, la fecha en que se le otorgó así como mantener un histórico de las clasificaciones que ha tenido y los motivos por los cuales ha cambiado de una a otra.

Este módulo incluye una sección llamada Información Diagnóstica Avanzada que permitirá registrar el resultado de las entrevistas que se les realizan a los privados de libertad en el transcurso de los primeros 60 días después de haberse iniciado las actividades previstas en el Plan Individual de Atención Integral. Las entrevistas de Trabajo Social, Criminología y Psicología que les son practicadas, están dirigidas a conocer información sobre la vida de los privados de libertad antes de haber cometido algún delito, así como las circunstancias que contribuyeron a la comisión del mismo, su ambiente familiar y evolución en la comunidad. [25]

Las funciones que realiza este módulo son muy parecidas a las del proceso de evaluación criminológica que se realiza en el Sistema Penitenciario cubano.

1.3.5. Sistema Automatizado para el Control del Recluso (SACORE).

En 1989 comienza la automatización de los Centros Penitenciarios cubanos a nivel nacional, con el diseño, la implementación y posteriormente la instauración del SACORE, que permite la gestión de los principales datos del sancionado y algunos aspectos del trabajo de la especialidad de Control Penal.

Este sistema fue implantado a partir de enero de 2003 y contiene como módulos principales Control Penal, Reeduación Penal y Orden Interior; los que se encuentran en explotación actualmente. Los módulos que conforman el SACORE poseen como Sistema Gestor de Base de Datos Oracle, y la programación de la aplicación se ejecuta sobre Developer. [18] [19]

Con la existencia de este sistema aún se almacena y manipula mucha información de la forma tradicional, en formato duro, o mediante herramientas que no ofrecen facilidades para el trabajo. En cuanto a la evaluación criminológica, este sistema no automatiza dicho proceso.

1.3.6. Resultados del estudio del arte.

Luego del estudio realizado se concluye que SACORE, sistema utilizado en la actualidad en los Centros Penitenciarios cubanos, no informatiza el proceso de evaluación criminológica; Lock&Track Corrections Information System y Elite Jails son sistemas privativos en los que no se hace referencia a ninguna funcionalidad que se encargue de obtener los datos relacionados con la personalidad del interno, sus condiciones de vida antes de ingresar a la prisión o algún tipo de información importante para este

proceso, además no se tiene acceso al código fuente de dichos sistemas por lo que la posibilidad de reutilización del mismo queda descartada.

El EPVNET es un sistema de software libre donde los procesos penitenciarios gestionados están basados en la legislación penal española por lo que no es compatible con los procesos penitenciarios cubanos, además el EPVNET no posee ninguna funcionalidad mediante la que se obtenga información que sea de utilidad para el proceso de evaluación criminológica.

El SIGEP, solución informática desarrollada para la República Bolivariana de Venezuela, posee el módulo de Clasificación que gestiona un proceso similar a la evaluación criminológica. Este sistema por la arquitectura con la que fue concebido no permite la reutilización del módulo pues el diseño no está basado en componentes ni en un conjunto de funcionalidades agrupables y extraíbles que puedan ser incorporadas en la implementación del SIDEPE. Además la evaluación criminológica del Sistema Penitenciario cubano recoge más información en sus entrevistas y las reglas por las que se rige la realización de dicho proceso son diferentes, por lo que el SIGEP no resulta de utilidad en ese aspecto.

Como resultado del análisis de las soluciones informáticas se concluye que solo el SIGEP informatiza un proceso similar a la evaluación criminológica realizada en los Centros Penitenciarios cubanos pero no puede ser reutilizado.

1.4. Metodología, tecnologías y herramientas utilizadas.

A continuación se realiza una breve descripción de la metodología, las tecnologías y las herramientas seleccionadas por el equipo de arquitectura del proyecto Prisiones Cuba, proceso que resulta fundamental cuando se desea realizar un software con la calidad necesaria y en el tiempo requerido.

1.4.1. Metodología de desarrollo.

RUP

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases que guiarán a los desarrolladores de sistemas a

elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. [8]

RUP² es una metodología de desarrollo tradicional y se caracteriza por ser:

- **Dirigido por casos de uso.** Los casos de uso describen los requisitos funcionales del sistema desde la perspectiva del usuario y se usan para determinar el alcance de cada iteración y el contenido de trabajo de cada persona del equipo de desarrollo. [21]
- **Centrado en la arquitectura.** La arquitectura permite ganar control sobre el proyecto para manejar su complejidad y controlar su integridad. Hace posible la reutilización a gran escala y provee una base para la gestión del proyecto. [21]
- **Iterativo e incremental.** Se divide en 4 fases: Inicio, Elaboración, Construcción y Transición, y cada una de ellas se divide en iteraciones. Cada iteración realizada añade funcionalidades al producto de software o mejora las existentes. Además en cada iteración se trabaja en un número de disciplinas haciendo énfasis en algunas de ellas. Las disciplinas propuestas por RUP son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, entre otras. [21]

Entre las ventajas que tiene la utilización de esta metodología se encuentra la evaluación en cada fase que permite la realización de cambios; funciona bien en proyectos grandes; es sencilla, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software y permite un seguimiento detallado en cada una de las fases de desarrollo.

Las desventajas de esta metodología son la gran cantidad de documentación que genera. [8]

Se define como metodología de desarrollo RUP debido a que es una metodología muy utilizada para el análisis, diseño, implementación, despliegue y documentación de sistemas orientados a objetos. Además por el tamaño y la duración aproximada que tendrá el proyecto Prisiones RUP es la metodología adecuada; pues su utilización se recomienda para proyectos de larga duración y en los que se vean

² Rational Unified Process

involucradas muchas personas. RUP será utilizada como guía a seguir durante el proceso de desarrollo del software y se documentará el mismo siguiendo la esta metodología.

1.4.2. Herramientas de Modelado

Visual Paradigm 5.0

Visual Paradigm 5.0 es una herramienta de modelado que se caracteriza por su flexibilidad ya que es multiplataforma, es decir, que tiene la capacidad de ejecutarse sobre diferentes sistemas operativos. [35]

Soporta todo el ciclo de vida del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Como lenguaje de modelado utiliza UML, ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones con calidad. Posee gran facilidad de uso y el entorno gráfico que brinda es agradable para los usuarios. Además tiene una buena interoperabilidad con otras herramientas CASE³ y con los principales IDE⁴. [35]

La principal desventaja de esta herramienta es que necesita licencia para ser utilizada.

Visual Paradigm 5.0 fue la herramienta de modelado de software elegida para representar los artefactos de la aplicación por las facilidades que brinda para la modelación de sistemas, además el equipo de desarrollo está familiarizado con el uso de esta herramienta.

Embarcadero ER\Studio 8.0

Embarcadero ER/Studio es una herramienta CASE para el modelado de datos que permite realizar tanto el diseño lógico como el diseño físico de los datos por separado. Esta herramienta permite crear y manejar diseños de bases de datos funcionales. Además de la transformación automática de un diseño lógico a un diseño físico para una plataforma específica de base de datos. Constituye una herramienta muy fácil de usar y con soporte para más de 30 de las bases de datos más importantes de la industria, entre ellas: Oracle. [10]

³ **Computer Aided Software Engineering**, Ingeniería de Software Asistida por Computadora.

⁴ **Integrated Development Environment**, Entorno de Desarrollo Integrado.

Se decidió seleccionar a ER\Studio porque es una herramienta comúnmente utilizada en el mundo del modelado de datos, especializada en esta función, con grandes facilidades para su utilización pues permite generar una base de datos partiendo del modelo físico y lógico; o el modelo físico y lógico partiendo de una base de datos ya existente. Además es compatible con un alto número de Gestores de Bases de Datos, incluyendo Oracle que es el que se utilizará para el almacenamiento y administración de los datos del sistema a desarrollar.

1.4.3. Sistema Gestor de Base de Datos Oracle 11g

Oracle es un Sistema Gestor de Base de Datos privativo que posee una gran robustez que garantiza la seguridad y la integridad de los datos. Algunas características que presenta este sistema son: facilita la administración de la base de datos, permite el almacenamiento de grandes volúmenes de datos, estabilidad, escalabilidad y flexibilidad porque puede ser utilizado en diversos sistemas operativos. [29]

El mayor inconveniente a la hora de usar Oracle es sin duda su precio, pues las licencias de esta herramienta son bastante costosas.

Como Sistema Gestor de Base de Datos se seleccionó Oracle por ser una herramienta comúnmente utilizada en el mundo, por la integridad y seguridad que proporciona a los datos almacenados, por las facilidades que brinda a los administradores a la hora de realizar las consultas a los datos contenidos en la base de datos y porque ofrece estabilidad a los datos por largos períodos de tiempo. Además entre los requisitos iniciales para la concepción de la aplicación el cliente pidió que Oracle fuera el gestor a utilizar.

1.4.4. Herramientas de desarrollo

NetBeans 7.0

NetBeans, es un entorno de desarrollo integrado o IDE, constituye un producto libre y gratuito sin restricciones de uso. Está bajo las licencias Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL).

NetBeans es una herramienta utilizada por programadores para escribir, compilar, depurar y ejecutar programas. Está hecho para el lenguaje de programación Java pero puede servir para otros lenguajes,

como Groovy. Posee un compilador que permite escribir a la vez que resalta la sintaxis, además es muy necesario a la hora de identificar errores y depurar el código. [30]

Como IDE se seleccionó NetBeans por ser libre y sin restricciones de uso, por facilitar el trabajo de los programadores a través de la detección de errores y de la depuración de código, y por soportar una gran cantidad de lenguajes de programación de código abierto, entre ellos Groovy.

SVN Tortoise

SVN Tortoise es un cliente de control de versiones que permite el versionado de las carpetas, pues implementa un sistema “virtual” de ficheros versionados que sigue la pista de los cambios en todos los árboles de directorios en el tiempo; además a través de SVN Tortoise se puede conocer el estado en el que se encuentra cada carpeta y fichero versionado pues se indica en pequeños iconos sobreimpresionados. Todos los comandos de Subversion están disponibles desde el menú contextual del explorador y TortoiseSVN añade su propio submenú allí. [34]

La principal importancia de esta herramienta radica en que varias personas pueden modificar y administrar el mismo conjunto de datos desde sus respectivos lugares de trabajo fomentando así la colaboración.

Se seleccionó SVN Tortoise por ser una herramienta libre, que no necesita muchos recursos para ser utilizada, que permite el control de las versiones de todos los componentes de la aplicación y la manipulación de los mismos a través de un amplio conjunto de opciones que proporciona este cliente. Además porque permite la realización del trabajo colaborativo entre todos los miembros del proyecto.

Contenedor Web (Apache Tomcat 6.0)

Es un servidor web con soporte para servlet y JSP, en la actualidad es usado como servidor web en entornos con alto nivel de tráfico y alta disponibilidad. Es un software de código abierto, multiplataforma y se encuentra bajo la licencia de Apache (Apache Software Licence). Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la Máquina Virtual Java. Desde su origen ha evolucionado hasta convertirse en un buen servidor en términos de eficiencia, funcionalidad y velocidad. [33]

Se seleccionó Apache Tomcat como contenedor web ya que puede ser utilizado en entornos que dispongan de la Máquina Virtual de Java, además es un software libre que puede ejecutarse sobre múltiples sistemas operativos, permite que el sistema desarrollado esté disponible por el tiempo que resulte necesario y porque puede ser integrado con el marco de trabajo seleccionado para el desarrollo del sistema.

1.4.5. Tecnologías y lenguajes

Groovy

Groovy es un lenguaje de programación orientado a objetos, dinámico y de código abierto hecho especialmente para la Máquina Virtual de Java. Como base para la confección de Groovy se usaron lenguajes tales como Java, Python y otros. Debido a esto usa una sintaxis muy parecida a la de Java, comparte el mismo modelo de objetos y de seguridad. También desde Groovy se puede acceder directamente a todas las API⁵s existentes en Java y el código escrito en Java es totalmente válido en Groovy. [16]

Se seleccionó Groovy porque posee una gran flexibilidad y escalabilidad. Por su dinamismo, integración con Java y sintaxis simple. Además este lenguaje es el que utiliza el marco de trabajo seleccionado para el desarrollo de la aplicación por lo que resulta imprescindible su utilización.

Framework⁶ Grails 1.3.7.

Grails es un framework para desarrollo web, de código abierto y que utiliza la Máquina Virtual de Java. Formando parte de este framework se encuentran principios tales como Convención sobre Configuración (Convention Over Configuration), que busca decrementar el número de decisiones que un desarrollador necesita hacer, ganando así en simplicidad pero no perdiendo flexibilidad por ello y No te Repitas (Don't Repeat Yourself), que promueve la reducción de la duplicación.

⁵ **A**pplication **P**rogramming **I**nterface, Interfaz de Programación de Aplicaciones.

⁶ Marco de trabajo

También tiene definidos patrones de diseño, entre ellos Inversión de Control (IoC) que define que las dependencias de un componente no deben ser gestionadas desde el propio componente con el objetivo de mantener los componentes lo más sencillos posible, incluyendo únicamente código que tenga relación con la lógica del negocio. Estas características en general proporcionan un entorno de desarrollo estandarizado y ocultan en gran parte los detalles de configuración. [2] [14]

Algunas de las características fundamentales de Grails son:

- *Alta productividad:* Grails posee tres características fundamentales que incrementan su productividad en comparación con los marcos de trabajo Java tradicionales: la poca existencia de XML, es un marco de trabajo preparado para funcionar desde el primer momento, y con funcionalidades disponibles a través de métodos dinámicos.
- *Persistencia:* GORM⁷ es el medio mediante el cual se persisten los datos del modelo de dominio a través de métodos dinámicos cuyas funcionalidades son similares a los DAO.[27]

Grails está construido sobre una serie de marcos de trabajo de código abierto tales como Hibernate utilizado para el mapeo objeto-relacional y para realizar consultas a la base de datos; y Spring para los flujos de trabajo e inyecciones de dependencias. [27]

Grails puede ser ampliado mediante plugins. Por su dinamismo y teniendo como lenguaje base Groovy, puede disminuir el ciclo de desarrollo ahorrando tiempo y agilizando el trabajo.

Arquitectura en Capas según Grails.

La arquitectura de Grails está conformada por 3 capas lógicas principales: Capa Web, Capa de Servicios, y Capa de Datos. Cada capa brinda sus utilidades a la capa inmediatamente superior y se sirve de las prestaciones que le proporciona la inmediatamente inferior. Esta arquitectura en capas por su diseño proporciona la facilidad de modificar cada capa todo lo posible sin infligir daños o alteraciones a la capa inmediata.

⁷ Grails Object Relational Mapping

Capa Web: En esta capa se encuentran las Vistas y la Lógica de Presentación, las cuales según la arquitectura que propone Grails estarán en la carpeta de código fuente. Las clases controladoras estarán ubicadas en la carpeta *controllers* y en la carpeta *view* las vistas. En la Lógica de Presentación se manejará todo el flujo web utilizando la implementación del patrón Modelo Vista Controlador (MVC) que brinda Grails, en el cual la lógica del negocio se separa de la presentación de la aplicación. Esto permite cambiar fácilmente el aspecto de la aplicación, sin modificar su comportamiento. [5]

Capa de Servicios: En la capa de servicios se encapsula toda la lógica del negocio en clases que son utilizadas por los controladores en la capa de presentación. Estas clases radicarán según la arquitectura propuesta por Grails en la carpeta de código fuente *services*. [5]

Capa de Datos: En la capa de datos se manejan las clases de dominio abstrayendo al programador del mecanismo de persistencia usado. Grails para evitar el trabajo directo con un gestor de base de datos y sus tablas, y permitir el trabajo con objetos en su lugar, utiliza Hibernate como una herramienta ORM⁸. Sin embargo dada la naturaleza dinámica de Grails y la adopción del principio Convención sobre Configuración, se crea una versión diferente de la implementación de ORM de Hibernate llamado GORM (Mapeo Objeto-Relacional de Grails). GORM consiste en un conjunto de métodos Groovy dinámicos que son los encargados de trabajar sobre Hibernate. [5]

⁸ Object Relational Mapping



Fig. 1 Arquitectura de Grails.

Se seleccionó Grails porque es un marco de trabajo que brinda múltiples ventajas para los desarrolladores como la simplificación de las configuraciones necesarias para garantizar el funcionamiento del mismo. La alta productividad que provee Grails hace que el proceso de desarrollo de software sea más rápido y puede ser extendido a través de plugins según las necesidades del sistema que se está implementando, característica que aumenta la facilidad de trabajo con el mismo.

JavaScript

JavaScript, lenguaje de script multiplataforma, es un lenguaje de programación del lado del cliente muy utilizado gracias a su compatibilidad con la mayoría de los navegadores modernos. La principal utilización de este lenguaje es integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y para realizar validaciones del lado del cliente. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez. JavaScript se ejecuta en la aplicación cliente al mismo tiempo que las sentencias van descargándose junto con el código HTML. [7] [28]

La principal utilización de JavaScript durante el desarrollo de la aplicación será en la realización de las validaciones, del lado del cliente, de los datos introducidos por los usuarios.

Dojo Toolkit 1.5

Dojo Toolkit es una biblioteca de JavaScript que contiene APIs y widgets (controles) que facilitan la implementación de páginas web y disminuyen el tiempo de desarrollo. Entre sus características principales se encuentran: está dividido en paquetes, manipulación de DOM y manejo de eventos. [1]

Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas, y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Proporciona una amplia gama de opciones en una sola biblioteca JavaScript y es compatible con navegadores como Internet Explorer, Mozilla Firefox, Google Chrome, entre otros. [1]

Dojo es seleccionado como complemento para Grails, específicamente para los componentes de interfaz gráfica, por las facilidades de manipulación de DOM y de eventos que provee esta biblioteca.

1.5. El proceso de Evaluación Criminológica.

La evaluación criminológica se utiliza en los Centros Penitenciarios para conocer el nivel de deterioro de la personalidad del interno y la disposición del mismo a la comisión de delitos o indisciplinas en prisión. Partiendo de ese estudio se orienta el trabajo especializado con cada uno de ellos lo que conlleva diseñar un plan de tratamiento individual y proponer una ubicación en régimen y colectivo.

La siguiente figura ilustra la realización del proceso de evaluación criminológica a los internos que se encuentran en tránsito por el Área de Ingreso como parte del ingreso a los Centros Penitenciarios.

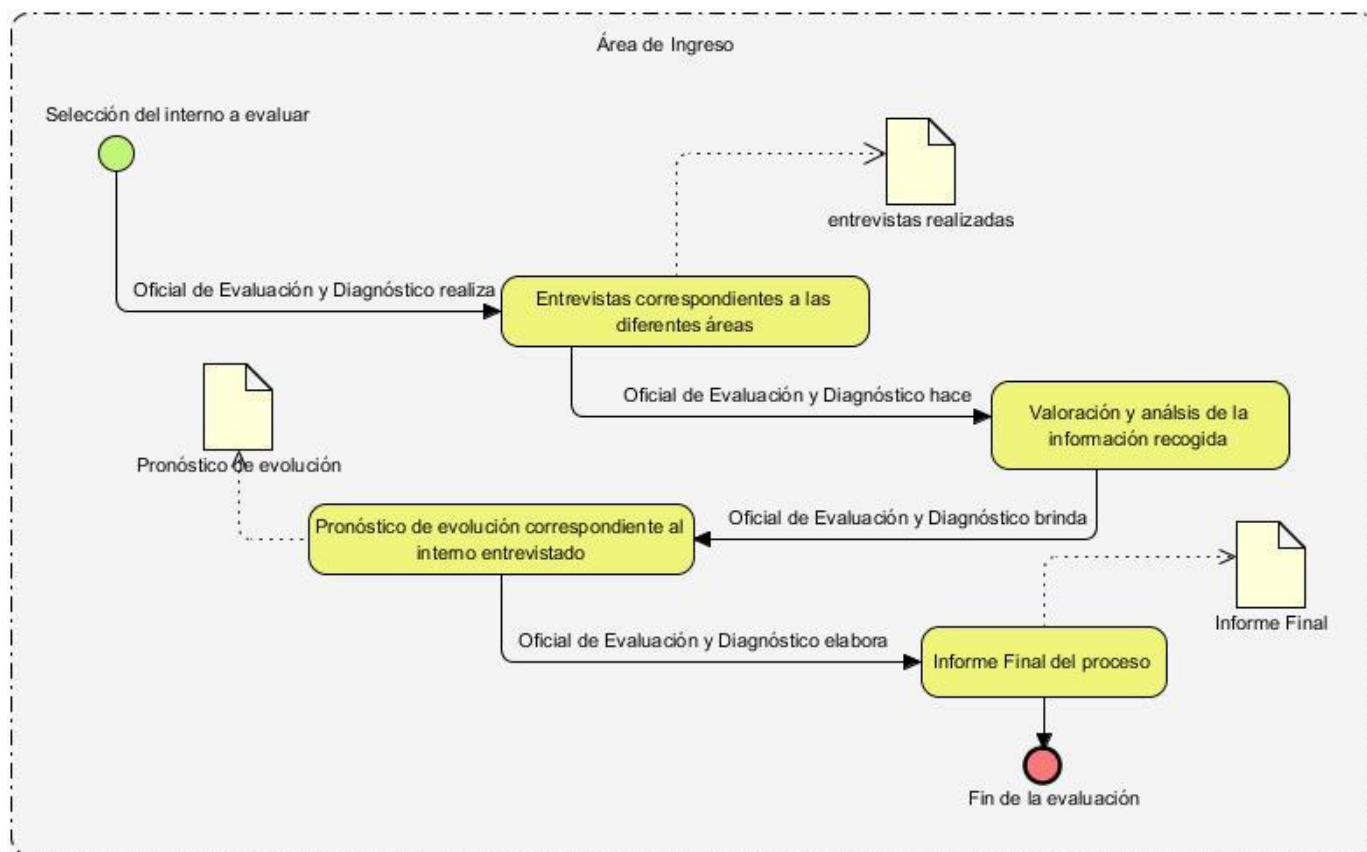


Fig. 2 Proceso de Evaluación Criminológica en el Área de Ingreso.

El proceso de evaluación criminológica se realiza durante la estancia en el Área de Ingreso del interno y comienza con la selección del individuo que se va a evaluar. El Oficial de Evaluación y Diagnóstico entrevista al interno recopilando toda la información posible de la forma de vida que ha mantenido antes de ingresar al centro. Luego de concluidas las entrevistas y con la previa valoración y análisis de las informaciones recogidas, se puede llegar a pronosticar la evolución de las características esenciales del interno. Con los elementos del estudio el Oficial de Evaluación y Diagnóstico elabora el informe final de evaluación.

1.6. Conclusiones Parciales

Durante el desarrollo de este capítulo se realizó un estudio de algunas soluciones informáticas para la gestión de los procesos penitenciarios, luego de esto se arribó a la conclusión de que no se cuenta con

una solución informática que gestione y satisfaga, de forma automatizada, las necesidades relacionadas con el proceso de evaluación criminológica del Sistema Penitenciario cubano, a partir de esto se hace necesario diseñar e implementar un sistema que sea capaz de gestionar completamente y de forma automatizada, el proceso de evaluación criminológica para el Sistema Penitenciario cubano.

También se analizaron y describieron, de forma breve, las herramientas y las tecnologías propuestas por el equipo de arquitectura del proyecto Prisiones Cuba para el diseño e implementación del módulo Evaluación Criminológica perteneciente al SIDE, y se describió el proceso de evaluación criminológica con el objetivo de aumentar la comprensión para el desarrollo del módulo.

CAPÍTULO II: DISEÑO DEL SISTEMA

2.1. Introducción

En el presente capítulo se describirá el diseño del sistema a implementar basándose en el análisis realizado previamente. Se describen las funcionalidades del sistema y se conforman los diferentes diagramas de clases y diagramas de secuencia correspondientes a los Casos de Uso que serán desarrollados, así como también el diseño de la base de datos.

2.2. Descripción del módulo Evaluación Criminológica

El módulo Evaluación Criminológica se encuentra dividido en 5 áreas fundamentales, el área Social, el área Personal, el área Criminológica, el área Impresiones Generales y el Pronóstico de Evolución. A cada una de estas áreas pertenecen un conjunto de entrevistas las cuales engloban los aspectos más importantes de la vida del individuo a entrevistar. Con los datos recopilados en las entrevistas y tomando como base esta información se brinda el pronóstico de evolución.

El encargado de realizar las entrevistas al interno y brindar el pronóstico de evolución en los Centros Penitenciarios es el Oficial de Evaluación y Diagnóstico.

A continuación se realiza una breve descripción de las funcionalidades organizadas por área.

Área	Funcionalidad	Descripción
Área Social	Gestionar información del Área Educativa.	Permite insertar la información relacionada con el área educativa, es decir la información relacionada con la etapa escolar del interno.
	Gestionar información del Área Familiar.	Permite insertar la información relacionada con el área familiar, es decir la información relacionada con el ambiente y la familia del interno.

	Gestionar información del Área Laboral.	Permite insertar la información relacionada con el área laboral, es decir todo lo que tiene relación con el desempeño laboral del interno.
	Gestionar información del Área de las Relaciones Interpersonales.	Permite insertar la información relacionada con el área de relaciones interpersonales, es decir todo lo que tiene relación con la interacción del interno con las demás personas.
Área Personal	Gestionar información del Área Cognitiva.	Permite insertar la información relacionada con el área cognitiva, es decir información relacionada con las cosas que influyeron en la forma de ser del interno.
	Gestionar información del Área Motivacional.	Permite insertar la información relacionada con área motivacional, es decir lo que motiva al interno.
	Gestionar información del Área Sexual y Matrimonial.	Permite insertar la información relacionada con el área sexual y matrimonial, es decir todo lo que tiene relación con el comportamiento sexual del interno.
	Gestionar información del Área de Antecedentes de Padecimiento.	Permite insertar la información relacionada con los antecedentes de padecimiento del interno.
	Gestionar Preguntas Generales	Permite insertar algunas informaciones generales relacionadas con el interno.
Área Criminológica	Gestionar información del Área Criminológica: Hábitos Tóxicos.	Permite insertar la información relacionada con el área criminológica, específicamente

		los hábitos tóxicos que presenta el interno.
	Gestionar información del Área Criminológica: Trayectoria Delictiva.	Permite insertar la información relacionada con el área criminológica, específicamente información relacionada con la trayectoria delictiva del interno.
	Gestionar información del Área Criminológica: Comportamiento en estado de reclusión.	Permite insertar la información relacionada con el área criminológica, específicamente información relacionada con el comportamiento del interno en su paso por instituciones penitenciarias.
Área Impresiones Generales	Gestionar información del Área Impresiones Generales.	Permite insertar la información relacionada con el área impresiones generales, específicamente información relacionada con la impresión del interno que tiene el oficial entrevistador.
Pronóstico de Evolución.	Gestionar información del Pronóstico de Evolución.	Permite insertar la información relacionada con la evaluación que brinda el psicólogo por cada una de las áreas registradas anteriormente.

Tabla 1 Descripción de las funcionalidades del módulo.

En cada funcionalidad del módulo se recogen los siguientes datos:

- *Área Cognitiva:* Esta entrevista recoge información sobre el comportamiento del interno, específicamente sobre la evolución de su forma de actuar a través de los años, para poder conocer si piensa las cosas antes de hacerlas, si reflexiona sobre las consecuencias de sus actos, etc.
- *Área Educativa:* Mediante esta entrevista se obtiene información sobre la educación del interno, desde su niñez hasta su adultez. Permite saber si asistió al círculo infantil, a la escuela y los

posibles problemas que tuvo en esta. De forma general se obtienen elementos que permiten determinar el comportamiento del interno y la correspondencia con la educación que tuvo.

- *Área Familiar:* Esta entrevista recoge información sobre el ambiente familiar en el que se desarrolló el interno. Desde con quién convivió durante su niñez, quién era la figura de autoridad en su casa, hasta si fue abandonado por sus padres. De forma general se obtienen elementos que permitan caracterizar las relaciones que posee el interno con sus familiares.
- *Área Laboral:* En esta entrevista se obtiene información sobre las experiencias laborales del interno. Permite conocer si le gustaba su trabajo, si se sentía bien haciéndolo o si prefiere cometer actos ilegales para obtener mayores beneficios económicos. En general recoge informaciones que permiten caracterizar la forma de ganarse la vida que prefiere el individuo.
- *Antecedentes de Padecimiento:* En esta entrevista se recoge información sobre los padecimientos de salud que pueda tener el interno y sus familiares, se buscan indicios de enfermedades mentales u otro tipo de enfermedad, para las cuales resulte necesario el tratamiento de un profesional.
- *Área Sexual y Matrimonial:* En esta entrevista se obtiene información relacionada con la sexualidad del interno. Se puede conocer sobre su primera experiencia sexual, las relaciones de pareja, los tipos de relaciones que prefiere, entre otras más. De forma general, informaciones que permitan conocer la importancia que le brinda el interno a las relaciones de pareja, a las relaciones con sus hijos y sus preferencias en este aspecto de su vida.
- *Área de Relaciones Interpersonales:* Esta entrevista recoge información sobre las relaciones que establece el interno con las personas que lo rodean. Se puede saber si tiene amigos, cuáles son las características de alguien considerado su amigo y si le resulta fácil relacionarse o no. En general, aspectos importantes sobre la forma de pensar que posee el individuo respecto a las relaciones afectivas.
- *Área Motivacional:* Esta entrevista recoge información sobre las cosas que motivan al interno, lo que le gustaría hacer y lo que él cree que necesita para lograrlo. De forma general, se obtiene

información sobre las cosas que lo atraen y le llaman la atención para enfocar el proceso de reformatión en ese sentido.

- *Preguntas Generales:* Esta entrevista recoge información sobre aspectos generales relacionados con algunas características personales de la vida del interno. Ejemplo de estos aspectos son: si es bueno mintiendo, si lo han acusado de manipulador, el momento de mayor felicidad y el de mayor tristeza en su vida, entre otros.
- *Hábitos Tóxicos:* Esta entrevista recoge información sobre los hábitos que posee el interno, dígase tabaco, café, bebidas alcohólicas y drogas. También permite conocer sobre los efectos que producen en él estos hábitos, en fin elementos para saber si el individuo tiene adicción por algo y cuál será el tratamiento que necesita.
- *Trayectoria Delictiva:* En esta entrevista se obtiene información sobre la etapa de la vida en la cual el interno comenzó a realizar actividades delictivas. Permite conocer si fue atendido por dichas actividades a temprana edad o nunca detectaron que las realizaba, que tipo de cosas hacía, etc. De esta forma se puede determinar qué tan antigua es la costumbre de delinquir en el individuo y si hay posibilidades de corregir su comportamiento.
- *Comportamiento en Estado de Reclusión:* Esta entrevista recoge la información relacionada con las estancias que ha tenido el individuo en la prisión. En caso de haber estado privado de libertad en ocasiones anteriores se puede conocer sobre las facilidades de las que disfrutó o si cometió indisciplinas en este período de tiempo. El objetivo principal es obtener conocimiento de su comportamiento en prisión y las medidas a tomar con el interno para evitar que realice acciones que comprometan la disciplina del centro.
- *Impresiones generales:* Esta entrevista recoge las principales impresiones que se llevó el oficial que entrevistó al interno para de esta forma saber si las respuestas están acordes a los gestos y formas de responder del individuo al que se le realizó la entrevista.
- *Pronóstico de Evolución:* Luego de realizadas todas las entrevistas pertinentes, se procede a realizar el pronóstico en el cual se recoge la valoración del Oficial de Evaluación y Diagnóstico

sobre las respuestas dadas por el interno en las entrevistas y brinda el pronóstico final sobre la personalidad del individuo.

Los datos recogidos por áreas son utilizados para la realización del trabajo especializado con los internos en sus respectivos centros. Al finalizar, las entrevistas y el pronóstico quedan almacenados en el Expediente Legal del interno.

2.3. Propósito del Diseño

Diseño puede definirse como: “... el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física.” [32]

El proceso de diseño del software constituye el final de la fase de elaboración y el comienzo de la fase de construcción, siendo un refinamiento del análisis que contribuye a una arquitectura estable y sólida. Esta fase posee una importancia vital para el posterior desarrollo del software, pues durante la etapa de diseño se modela el sistema para que cumpla con todos los requisitos que necesita el cliente, las restricciones relacionadas con los lenguajes de programación, sistemas operativos, tecnologías de interfaz de usuario, entre otros.

2.4. Modelo de Datos.

El Modelo de Datos describe la representación lógica y física de los datos que persistirán, este modelo permite definir la estructura que tendrá la base de datos del sistema.

La figura siguiente muestra el modelo lógico de los datos donde se representan las entidades pertenecientes al Caso de Uso (CU) Gestionar Información del Área Laboral.

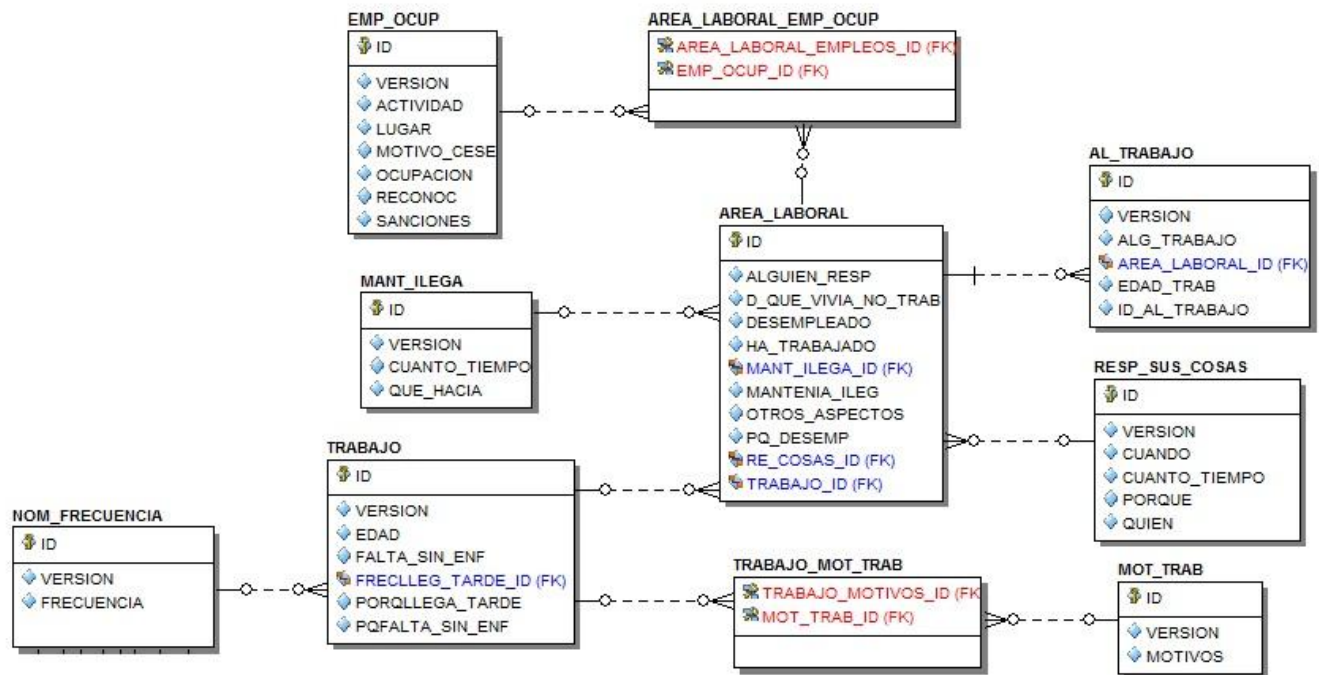


Fig. 3 Modelo de Datos Área Laboral.

2.5. Diagrama de Clases del Diseño

Los diagramas de clases del diseño constituyen la primera representación concreta del sistema. En cada uno de esos diagramas se muestran las vistas con los formularios que las componen, así como también las clases JavaScript correspondientes a cada vista. En estos diagramas también se representan las clases controladoras, clases servicios y clases del dominio necesarias para la posterior implementación del sistema.

La siguiente figura muestra el diagrama de clases del diseño del CU Gestionar Información del Área Laboral.

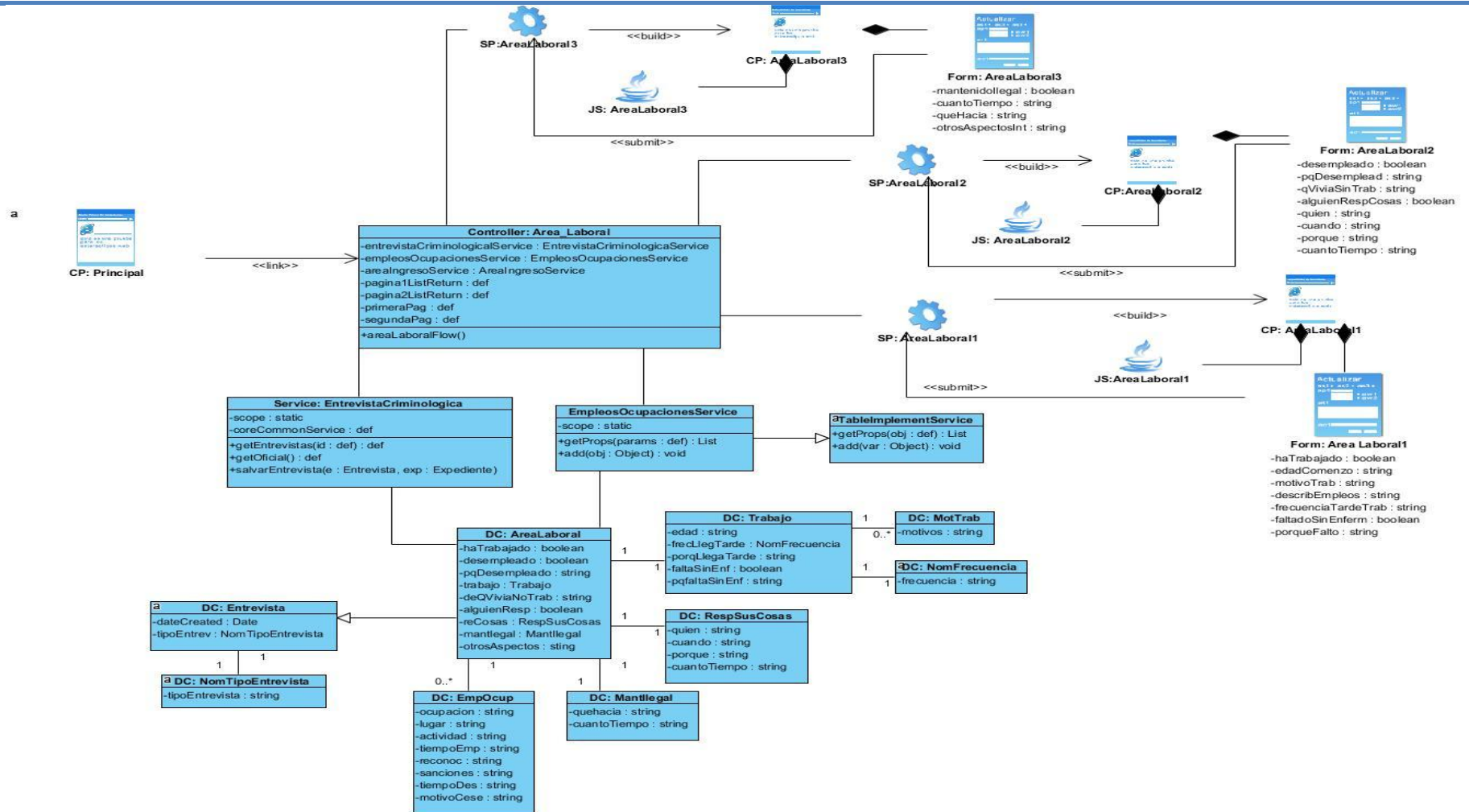


Fig. 4 Diagrama de Clases del Diseño del CU Gestionar Información del Área Laboral.

2.6. Patrones de diseño utilizados

Los patrones de diseño son soluciones estándares para problemas comunes, estos representan un conjunto de reglas, probadas y documentadas anteriormente, que describen cómo afrontar tareas y solucionar problemas que surgen durante el desarrollo de software.

2.6.1. Modelo Vista Controlador (MVC)

El patrón MVC propone que los componentes de un sistema estén organizados en 3 capas distintas según su misión dentro del mismo. Para el diseño del módulo Evaluación Criminológica se utilizó este patrón, el cual es muy popular en el mundo del desarrollo de aplicaciones web.

El patrón MVC abarca la Capa Web y la Capa de Datos dentro de la arquitectura de Grails, y se encuentra estructurado en las 3 capas siguientes.

- **Modelo:** Contiene los componentes que representan los datos manejados por la aplicación, es decir, a esta capa pertenecen las clases del dominio.
- **Vista:** Los componentes de esta capa son los encargados de la representación gráfica del sistema, mostrar las diversas acciones disponibles, así como también el estado actual de los datos. A esta capa pertenecen las vistas, los archivos JavaScript y los archivos de estilo.
- **Controlador:** Contiene los componentes que reciben las peticiones de los usuarios y gestionan el cumplimiento de dichas peticiones. A esta capa pertenecen las clases controladoras.

2.6.2. Singleton

Por defecto las instancias de los servicios tienen una vida única durante la ejecución de la aplicación, lo que permite que se ponga de manifiesto el patrón Singleton, este patrón pertenece a los patrones GoF⁹. Este criterio es muy útil en la mayoría de los casos en los que se necesite almacenar información de larga vida, que sea accesible desde cualquier lugar y momento, pero tiene como inconveniente que no se puede guardar información que sea “privada” pues terceros podrían acceder a ella.

⁹ Gang of Four o Banda de Cuatro

En el módulo Evaluación Criminológica se utiliza el patrón Singleton y para solucionar el inconveniente de la privacidad se utiliza el concepto de ámbito de sesión (session), petición (request) o flujo (flow). Los cuales garantizan la creación de una instancia del servicio según el ámbito. Para definir un servicio de sesión se declara el atributo estático "scope" como se muestra a continuación. En este ejemplo se ha definido que por cada sesión de usuario se cree una instancia de la clase EntrevistaCriminologicaService porque dicho servicio es único para cada sesión pues se manipulan datos a los que no se puede acceder desde otras sesiones.

Ejemplo:

```
class EntrevistaCriminologicaService  
  
{  
    static scope = "session"  
  
    //código de la clase  
  
}
```

2.6.3. Patrón Experto

El patrón Experto, pertenece a los patrones GRASP¹⁰, propone que la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Se evidencia el uso de este patrón en la asignación de responsabilidades a las clases que contienen la información necesaria para el almacenamiento en la base de datos.

2.6.4. Patrón Controlador

Este patrón, pertenece a los patrones GRASP, propone que los controladores solo supervisen la realización de las actividades y deleguen en otros objetos el trabajo que se desea hacer. Se evidencia el

¹⁰ **General Responsibility Assignment Software Patterns** o Patrones de Principios Generales para Asignar Responsabilidades

uso de este patrón en las clases controladoras pues estas manejan los mensajes de los eventos del sistema y coordinan las actividades que deben realizarse.

2.7. Diagramas de Interacción. Diagrama de Secuencia.

Un diagrama de interacción muestra la interacción entre objetos, es decir, contiene un conjunto de objetos y sus relaciones, así como también los mensajes que se pueden enviar entre ellos, destacando la secuencia temporal de dichos mensajes.

En la siguiente figura se muestra el diagrama de secuencia perteneciente al CU Gestionar Información del Área Laboral.

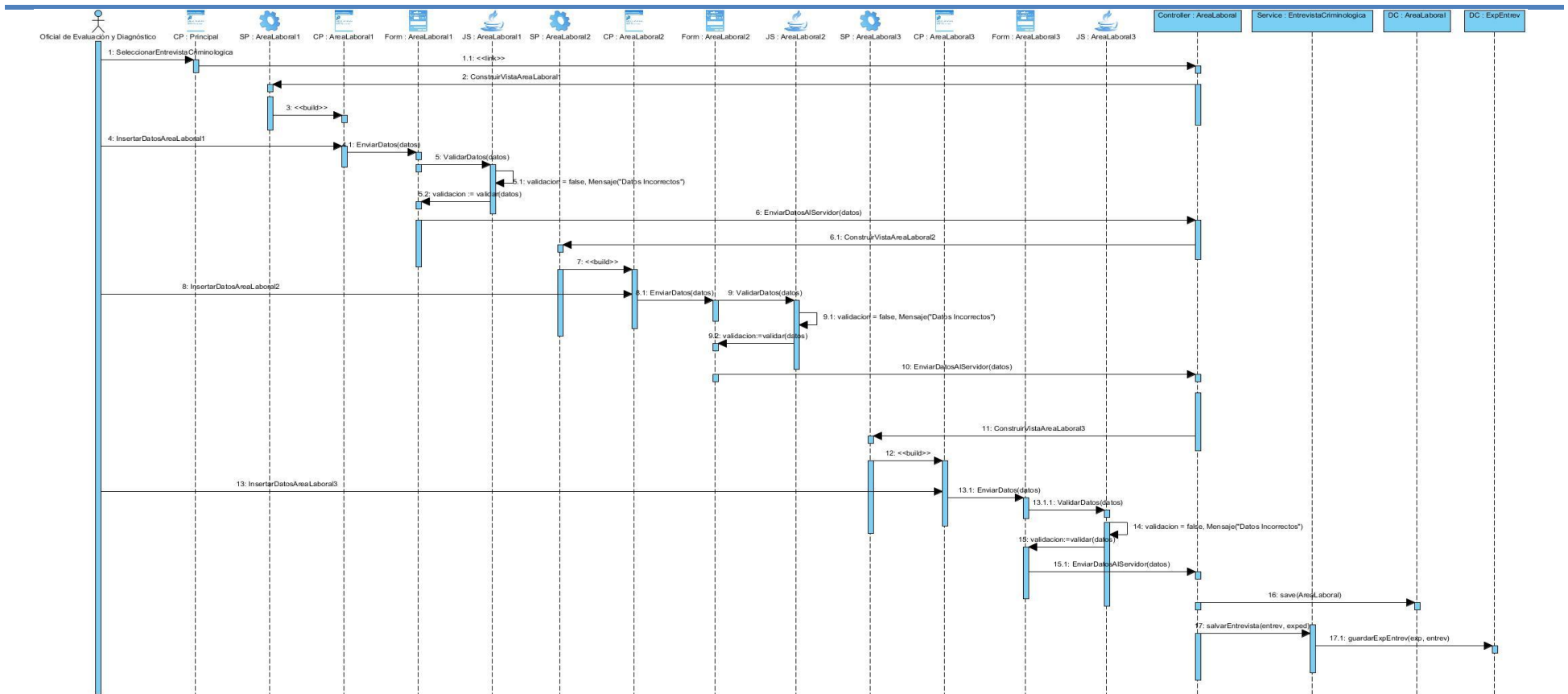


Fig. 5 Diagrama de Secuencia perteneciente a Registrar Información del Área Laboral.

Para ver los diagramas pertenecientes a los demás CU: *Ver anexos*.

2.8. Flujo Web (Web Flow)

Los flujos web (web flows) constituyen una conversación entre el navegador y la aplicación web que se extiende a lo largo de varias peticiones manteniendo un información entre ellas. Se puede definir el estado inicial y el final, así como gestionar las transiciones entre estados. [26]

El estado inicial es el punto de inicio del flujo web. Las transiciones son indicaciones de cambio de estado que se realizan dentro de cada estado en particular, una transición se define por 3 parámetros: origen, destino y evento.

El origen es el estado desde el que se define la transacción. El evento es el resultado de la ejecución de un estado. Cada estado finaliza en un estado o en otro en función del resultado de la ejecución o de la entrada que proporcione el usuario. El destino es una referencia al estado al que se tiene que pasar posteriormente. [26]

Grails permite la creación de flujos web basado en Spring Web Flow [26] la convención adoptada en Grails es que cualquier acción cuyo nombre termine en *Flow* definirá un flujo web.

A continuación se muestra el diagrama de estados que representa el flujo web diseñado para implementar el Área Laboral.

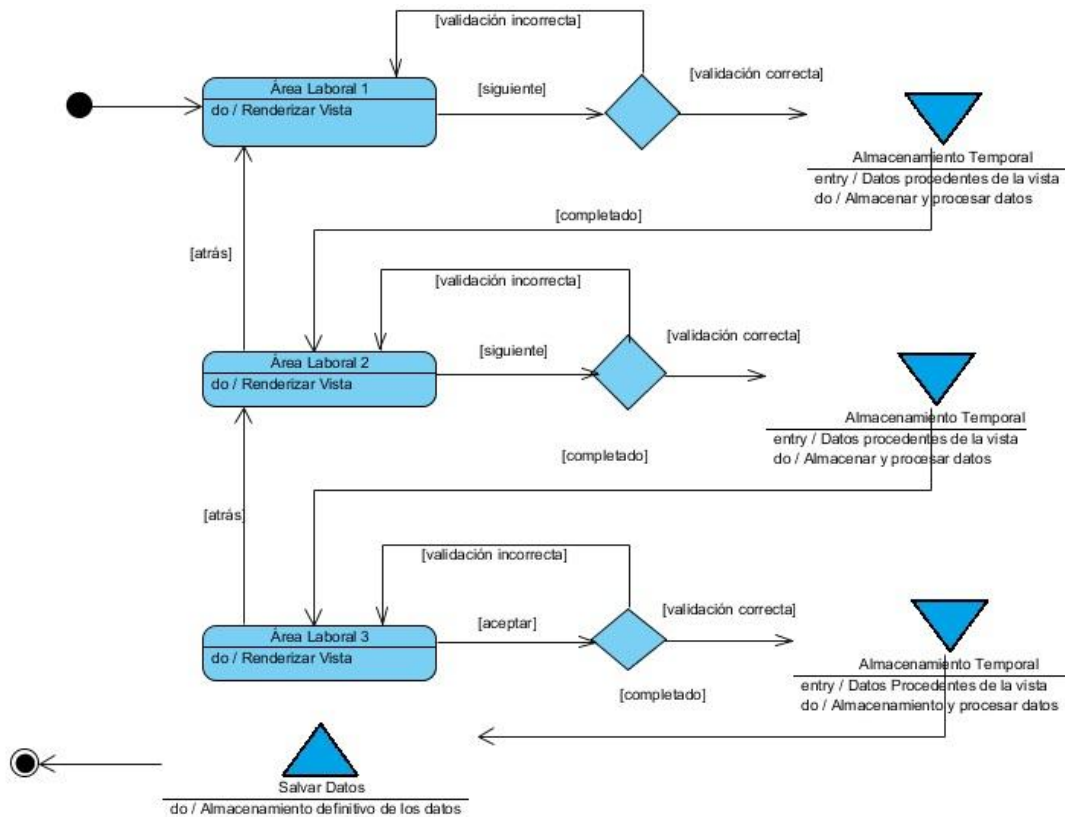


Fig. 6 Diseño del Flujo Web para el CU Gestionar Información del Área Laboral.

En la figura anterior se representa la secuencia que debe seguir el flujo web diseñado para el CU Gestionar Información del Área Laboral. El flujo comienza cuando es mostrada la primera página del Área Laboral para introducir los datos correspondientes, seguido de una transición producida por la acción del usuario al presionar el botón *siguiente*. Transición esta que tiene como origen el estado *Área Laboral 1* y como destino *Almacenamiento Temporal*. En el momento que se produce la transición los datos son validados, en caso de existir algún error se regresa al estado inicial y en el caso contrario se llega al estado destino. En el estado *Almacenamiento Temporal* los datos introducidos son almacenados en el flujo. Luego de completar esta acción se produce otra transición que tiene como origen el estado *Almacenamiento Temporal* y como destino *Área Laboral 2*, esta vez la transición está dada por el completamiento del almacenamiento de los datos. En el estado *Área Laboral 2* el usuario puede presionar el botón *atrás*, acción con la cual se producirá una transición que llevará al sistema al estado *Área Laboral*

1, o puede presionar el botón *siguiente* lo que permitirá que el sistema se comporte de la misma manera que cuando se presiona el botón siguiente en el estado *Área Laboral 1*. Así se recorren todos los estados sucesivamente hasta llegar al estado *Salvar Datos* en el cual se persisten todos los datos que se encontraban almacenados en el flujo y se llega al estado final, con lo que finaliza el flujo web.

Conclusiones Parciales

Al concluir el presente capítulo se han generado artefactos tales como los Diagramas de Clases del Diseño, los Diagramas de Interacción, el Diagrama de Flujo Web y el Modelo de Datos perteneciente al módulo Evaluación Criminológica del SIDE P. También se hizo referencia a los patrones utilizados para el diseño del módulo y se realizó una breve descripción del módulo y de las funcionalidades que este contiene.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA

3.1. Introducción

La implementación del sistema comienza después de haber terminado los flujos de trabajo de análisis y diseño. Con los artefactos obtenidos se inicia la implementación, fase que es de vital importancia para el desarrollo de cualquier sistema. En el capítulo se presenta el diagrama de despliegue y el diagrama de componentes correspondientes al módulo.

Además se explica, usando ejemplos, la implementación de las clases del dominio, controladores y servicios. También se realizarán las pruebas correspondientes para comprobar que el producto posee la calidad requerida.

3.2. Implementación

Durante la fase de implementación se implementan las clases del diseño y se organizan en componentes, se ubican los componentes en nodos y se define la estructura en capas de la aplicación.

Para la implementación del módulo Evaluación Criminológica se realiza el diagrama de componentes y el de despliegue.

3.2.1. Implementación de las clases del Dominio

Las clases del dominio se encuentran ubicadas en la carpeta grails-app/domain, formando así el modelo de datos de la aplicación.

Grails utiliza GORM, de esta forma controla el ciclo de vida de las entidades y proporciona métodos dinámicos que facilitan la realización de búsquedas. GORM se encarga de relacionar entidades con las tablas de la base de datos y las propiedades de dichas entidades con los atributos de las tablas en la base de datos.

A continuación se muestra un ejemplo de la implementación de las clases del dominio pertenecientes al módulo Evaluación Criminológica.


```

1 package sidep.areaingreso.domain.evaluacioncriminologica.arealaboral
2 import sidep.areaingreso.domain.evaluacioncriminologica.Entrevista
3 /**
4  * Entidad para el area laboral de la entrevista social
5  */
6 class AreaLaboral extends Entrevista {
7
8     Boolean haTrabajado
9     Boolean desempleado
10    String pqDesemp
11    Trabajo trabajo
12    String dQueViviaNoTrab
13    Boolean alguienResp
14    RespSusCosas reCosas
15    Boolean mantIlegalleg
16    MantIlega mantIlega
17    String otrosAspectos
18
19    static hasMany = [empleos: EmpOcup]
20    static mapping = {
21        table 'evalAreaLab'
22    }
23    static constraints = {
24        trabajo(nullable: true)
25        pqDesemp(nullable: true)
26        reCosas(nullable: true)
27        mantIlega(nullable: true)
28        otrosAspectos(nullable: true)
29        dQueViviaNoTrab(nullable: true)
30    }
31 }

```

Fig. 7 Implementación de la Clase del Dominio AreaLaboral.

La clase *AreaLaboral.groovy* pertenece a la implementación del CU Gestionar Información del Área Laboral y contiene información relacionada con las experiencias laborales del individuo, su comportamiento en los centros de trabajo, la forma de sustento que tiene y si es legal, entre otros aspectos. Algunos de los atributos que componen la clase son los siguientes:

- **haTrabajado**: atributo booleano que permite conocer si el interno ha trabajado alguna vez o no.
- **desempleado**: atributo booleano que permite conocer si el interno ha estado desempleado alguna vez o no.
- **trabajo**: atributo de tipo *Trabajo* que permite conocer la edad en que comenzó a trabajar el individuo, si llegaba tarde, por qué llegaba tarde, si faltaba al trabajo sin motivo y por qué lo hacía.
- **alguienResp**: atributo booleano que permite conocer si alguien era responsable de sus cosas después de la mayoría de edad o no.

- **otrosAspectos**: atributo String que permite conocer otros aspectos de interés relacionados con el área laboral y que no hayan estado reflejados en las preguntas anteriores.

Además la clase *AreaLaboral.groovy* se relaciona con la clase *EmpOcup.groovy* que permite conocer detalles sobre los empleos u ocupaciones que ha tenido el individuo.

Los atributos **trabajo y otrosAspectos** pueden ser nulos en caso de que el individuo responda que no en alguna de las preguntas realizadas.

3.2.2. Implementación de las clases Controladoras

Las clases controladoras se encuentran ubicadas en la carpeta *grails-app/controllers*, y son las responsables de recibir las solicitudes del usuario, gestionar la aplicación de la lógica del negocio y decidir la vista que se debe mostrar a continuación.

A continuación se muestra un ejemplo de la implementación de las clases controladoras pertenecientes al módulo Evaluación Criminológica.

```
1 package sidep.areaingreso.controllers.evaluacioncriminologica.entrevistas.criminologica.social
2 import ...
6
7 class AreaLaboralController {
8
9     def entrevistaCriminologicaService
10    def empleosOcupacionesService
11    def areaIngresoService
12    def pagina1ListReturn = [:]
13    def pagina2ListReturn = [:]
14    def primeraPag = [:]
15    def segundaPag = [:]
16
17    def index = { redirect(action: "areaLaboral") }
18
19    def addEmpleosOcupaciones = {...}
23    def delEmpleosOcupaciones = {...}
26
27    def areaLaboralFlow = {
28        inicial {
29            render view: "/evaluacioncriminologica/entrevistas/criminologica/social/areaLaboral1"
30            on("siguiente") {
31
32                if (params.preguntalEnAlgunMomentoHaTrabajado == "si") {
33
34                    flow.haTrabajado = true
35                    primeraPag.trabajadoSi = true
36                    def oTrabajo = new Trabajo()
37
```

Fig. 8 Implementación clase AreaLaboralController.

La clase *AreaLaboralController.groovy* pertenece a la implementación del CU Gestionar Información del Área Laboral y en ella se manipulan las peticiones que realizan los usuarios a través de las vistas. En este caso se recogen los datos que introduce el usuario utilizando el flujo web, moviéndose así por diferentes estados hasta llegar al estado final en el cual se almacenarán los datos definitivamente.

En esta clase se tienen instancias de *EntrevistaCriminologicaService*, servicio que se encarga de guardar las entrevistas así como también el expediente con el que se asocia la misma; *EmpleosOcupacionesService*, servicio que mantiene los datos de los empleos que ha tenido el interno recogidos de las vistas en memoria hasta el momento en el que se almacenan definitivamente y *AreaIngresoService*, servicio a través del cual se obtiene el expediente del interno que está siendo entrevistado.

Otros atributos presentes en la clase son:

- **pagina1ListReturn:** mapa que almacenará los datos procedentes de la página 1 de la forma *nombre: valor*. Estos datos serán utilizados en el momento del flujo en que el usuario decida regresar a la página 1 para llenar los campos de dicha página.
- **pagina2ListReturn:** mapa que almacenará los datos procedentes de la página 2 de la forma *nombre: valor*. Estos datos serán utilizados en el momento del flujo en que el usuario decida regresar a la página 2 para llenar los campos de dicha página.
- **primeraPag:** mapa que almacenará los datos booleanos procedentes de la página 1.
- **segundaPag:** mapa que almacenará los datos booleanos procedentes de la página 2.

3.2.3. Implementación de las clases Servicios

Según la convención que sigue Grails, un servicio es una clase cuyo nombre termina en *Service* y se encuentran ubicados en la carpeta *grails-app/services*. En tiempo de ejecución Grails usará a Spring para hacer que todas las clases que declaren variables con el mismo nombre del servicio tengan una instancia a su disposición.

A continuación se muestra un ejemplo de la implementación de las clases servicios pertenecientes al módulo Evaluación Criminológica.

```
1 package sidep.areaingreso.services.evaluacioncriminologica.entrevistas.criminologica.arealaboral
2
3 import sidep.core.service.table.TableImplementsService
4
5 class EmpleosOcupacionesService extends TableImplementsService {
6     static scope = "session"
7
8     public List getProps(def params) {
9         ["Ocupacion", "Lugar", "Actividad", "Tiempo_Empleo", "Reconocimientos", "Sanciones", "Tiempo_Desempleo",
10          "Motivo_Cese"]
11     }
12
13     @Override
14     public void add(Object var) {
15         def n = [id: idElement++, Ocupacion: var.ocupacion, Lugar: var.lugar, Actividad: var.actividad,
16                 Tiempo_Empleo: var.tiempo, Reconocimientos: var.reconocimientos, Sanciones: var.sanciones,
17                 Tiempo_Desempleo: var.tiempoDesempleo, Motivo_Cese: var.motivoCese]
18         items << n
19     }
20
21 }
```

Fig. 9 Implementación de la clase `EmpleosOcupacionesService`

La clase `EmpleosOcupacionesService` es uno de los servicios utilizados en la implementación del CU Gestionar Información del Área Laboral, el mismo hereda de la clase `TableImplementsService`. En el servicio se redefinen los métodos `getProps()` y `add()`, adaptándolos así a las características y a las necesidades de la clase.

En `EmpleosOcupacionesService` se define una variable estática **scope** con el objetivo de restringir el ámbito del servicio a la sesión activa en el momento.

3.2.4. Creación de Entidades

Todas las clases del dominio que se encuentran en la carpeta `grails-app/doman` serán interpretadas como entidades. Luego de definir las propiedades de esas clases serán generadas las tablas correspondientes en la base de datos con los campos necesarios para guardar cada propiedad definida y con los métodos necesarios para manipular la entidad.

3.2.5. Operaciones sobre el Modelo de Datos

Métodos de Actualización: Las entidades de la aplicación poseen métodos dinámicos que facilitan el trabajo con las mismas.

- **save():** Para insertar el registro en la base de datos o para actualizarlo si ya existe. Usando el argumento *flush:true* los datos se envían a la base de datos en el mismo momento en que se invoca el método `save()`.
- **delete():** Para eliminar un registro.

Dynamic Finders (localizadores dinámicos)

En algunas clases se usaron los métodos `findBy()` y `findAllBy()`. La diferencia entre estos métodos es que el primero solo devuelve la primera instancia que cumpla la condición de búsqueda, mientras que el segundo devolverá una lista con todos los resultados que correspondan. Estos localizadores resultan muy poderosos a la hora de realizar consultas básicas sobre el modelo de datos, pero no permiten hacer búsquedas avanzadas que incluyan muchos campos y comparaciones complejas. Para esto se recomienda construir consultas propias utilizando `Criteria`.

3.2.6. Internacionalización

Internacionalizar una aplicación significa diseñarla de forma tal que los mensajes transmitidos a los usuarios puedan ser visualizados en la interfaz en distintos idiomas sin que sea necesario modificar el código fuente.

Con Grails existe soporte para la internacionalización por medio de archivos de recursos almacenados en la carpeta `grails-app/i18n`. Los archivos de recursos son ficheros de propiedades en los que se guardan las distintas versiones de cada mensaje en los idiomas deseados. Cada idioma está definido en un fichero distinto que incluye en su nombre el código del idioma, por ejemplo: `messages_es_ar.properties` para los mensajes en español de Argentina; `messages_en.properties` para los mensajes en inglés y `messages_es.properties` para los mensajes en español internacional.

A continuación se muestra la utilización del fichero de internacionalización, en este ejemplo se utiliza este fichero para externalizar los textos en la interfaz de usuario de la aplicación.

```

64 # Entrevista criminologica
65 # Pronostico de evolucion
66 domain.pronosticoEvolucion.label=Pronóstico de evolución
67 domain.contenidoPronostico.label=Respuesta
68 domain.pronosticoInterno.label=Pronóstico correspondiente al interno
69

```

Fig. 10 Fichero de internacionalización para el módulo Evaluación Criminológica.

Luego de tener especificadas las claves en el fichero de internacionalización, se define en la interfaz de usuario una etiqueta en la que se hace referencia a una de las claves contenidas en el fichero, esta etiqueta mostrará el valor que tiene asignado la clave utilizada.

Ejemplo: `<g:message code="domain.pronosticoEvolucion.label" />`

```

24 <form id="pronosticoEvolucionForm-id" action="${createLink(action: "save")}">
25   <s:title code="domain.pronosticoEvolucion.label" />
26   <s:div class="formpanel1">
27     <ul style="padding-left: 20px;"...>
47   </s:div>
48   <s:clear/>
49   <s:title code="domain.pronosticoInterno.label"/>
50   <s:select name="tipo_id" nom="NomTipoPronostico" />
51   <s:clear/>
52   <hr/>
53   <s:buttonBar>
54     <s:button name="aceptar"/>
55     <s:button name="cancelar" type="reset"/>
56   </s:buttonBar>
57   <s:clear/>
58 </form>

```

Fig. 11 Utilización del fichero de internacionalización en las vistas.

3.2.7. Diagrama de componentes

El diagrama de componentes es utilizado para modelar los componentes de un sistema, incluyendo los artefactos que implementan dichos componentes y las dependencias entre ellos.

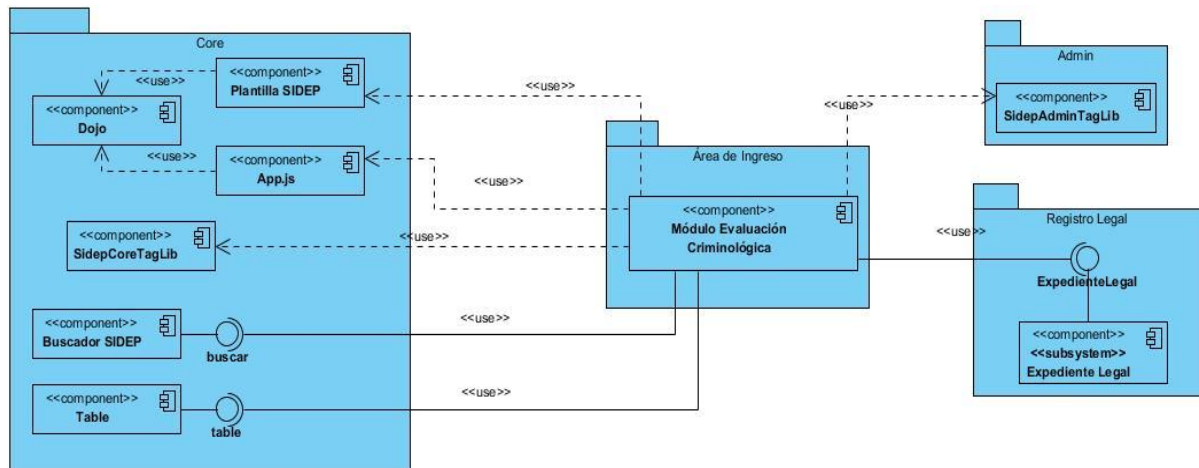


Fig. 12 Relaciones del módulo Evaluación Criminológica con otros componentes del sistema.

La figura anterior muestra las relaciones del módulo Evaluación Criminológica con otros componentes del SIDEP. El módulo hace uso de: *Plantilla SIDEP* que ofrece el estilo a las vistas, lo que brinda uniformidad a todas las interfaces de usuario de la aplicación; *App.js* es un archivo JavaScript común para todo el SIDEP encargado de realizar validaciones y lanzar eventos de carácter universal en la aplicación; *SidepCoreTagLib* librería de etiquetas utilizada a la hora de confeccionar las vistas; *Buscador SIDEP* utilizado para realizar búsquedas de objetos en la aplicación; *Table* componente que permite la utilización de tablas dinámicas en las vistas; *SidepAdminTagLib* que permite la seguridad de los datos del sistema y el *Expediente Legal* componente que contiene datos esenciales para el trabajo en las entrevistas del módulo Evaluación Criminológica.

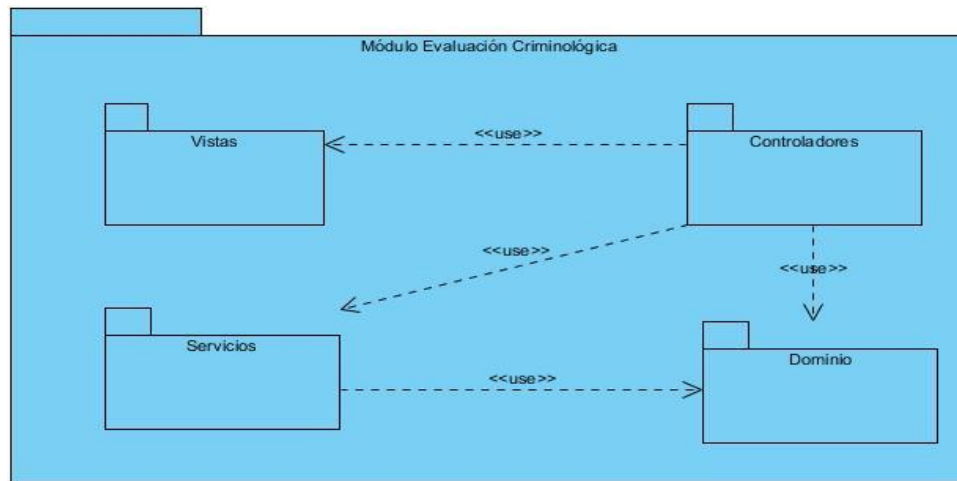


Fig. 13 Organización de los componentes del Módulo Evaluación Criminológica.

La figura anterior representa el diagrama de paquetes del módulo Evaluación Criminológica. Seguidamente se explica de forma breve el contenido de los paquetes utilizados.

Paquete Vistas

El paquete Vistas pertenece a la Capa Web de la arquitectura, dentro se encuentran los ficheros de tipo GSP (las vistas) que representan la interfaz de usuario del sistema.

Paquete Controladores

El paquete Controladores pertenece a la Capa Web de la arquitectura, dentro se encuentran las clases controladoras que son las encargadas de atender las peticiones que proceden de las vistas, así como también de brindar los datos necesarios para satisfacer al usuario.

Paquete Servicios

El paquete Servicios pertenece a la Capa Servicios de la arquitectura. Dentro se encuentran los servicios, que son las clases encargadas de almacenar, extraer y consultar los datos para responder a los pedidos de los controladores.

Paquete Dominio

El paquete Dominio pertenece la Capa de Datos de la arquitectura. Dentro se encuentran las clases del dominio, las que se corresponden con las entidades, donde se guarda toda la información de la aplicación.

3.2.8. Diagrama de despliegue

El Diagrama de Despliegue describe la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de los nodos que conforma la estructura.

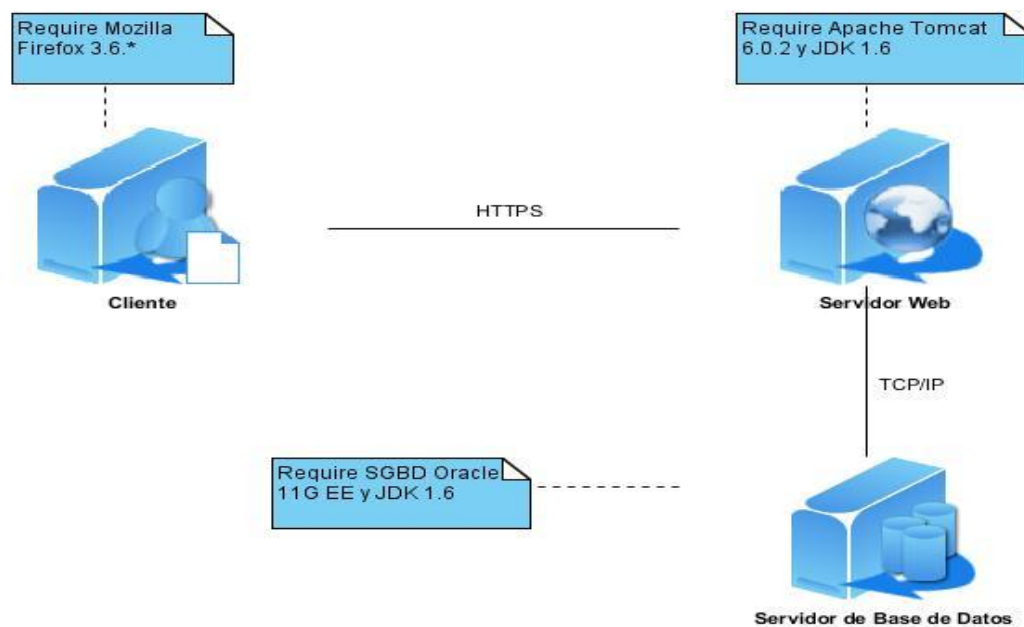


Fig. 14 Diagrama de despliegue Módulo Evaluación Criminológica.

Para el correcto despliegue del módulo Evaluación Criminológica se necesita un local seguro, donde estará ubicado el Servidor Web y el de Base Datos. El Oficial de Evaluación y Diagnóstico hará uso de la aplicación a través de clientes ligeros.

Los requisitos a cumplir son los siguientes:

➤ **Servidor de Base de Datos**

- Microprocesador: 4 núcleos, 3 GHz

- RAM: 4 GB
- Espacio necesario para la instalación: 2 GB
- Espacio libre: 1 TB
- Sistema Operativo: Linux Suse Enterprise Server 10.0
- Plataforma Java: JDK 1.6
- SGBD¹¹: Oracle 11g

➤ **Servidor Web**

- Microprocesador: 4 núcleos, 3 GHz
- RAM: 4 GB
- Espacio necesario para la instalación: 250 MB
- Espacio libre: 250 GB
- Sistema Operativo: Linux Suse Enterprise Server 10.0
- Plataforma Java: JDK 1.6
- Contenedor web: Apache Tomcat 6.0.2

➤ **PC clientes**

- RAM: 512 MB
- Sistema Operativo: Microsoft Windows XP
- Navegador web: Mozilla Firefox 3.6.*

¹¹ Sistema Gestor de Base de Datos

3.3. Pruebas

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un sistema informático. Básicamente es una fase en el desarrollo de software, consistente en probar las aplicaciones construidas. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto a las especificaciones iniciales del sistema. [13]

3.3.1. Método de pruebas: Pruebas de Caja Negra

Se denominan pruebas de caja negra a un tipo de prueba de software que se realizan sobre la interfaz del software, tiene como objetivo verificar que la entrada se acepta de forma adecuada y que se produce un resultado correcto, sin tener en cuenta su funcionamiento interno. En estos casos el probador proporciona las entradas y estudia las salidas para ver si son o no las esperadas. [13]

3.3.2. Técnica de pruebas: Diseño de casos de prueba

Casos de prueba

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución, resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. La entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final. [23]

Los casos de pruebas deben verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificación de los requerimientos.
- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

Un caso de prueba se diseña según las funcionalidades descritas en los casos de uso. Este diseño se elabora previo a la realización de las pruebas funcionales de la aplicación. Cada planilla de caso de

prueba recoge la especificación de un caso de uso, dividido en secciones y escenarios, para hacer más fructífera la ejecución de las pruebas.

3.3.3. Nivel de Pruebas

Existen diversos niveles de pruebas entre ellas: las pruebas de unidad, las pruebas de integración, las pruebas de validación y las pruebas de sistema. La siguiente figura ilustra el nivel de realización de cada una de ellas. [23]

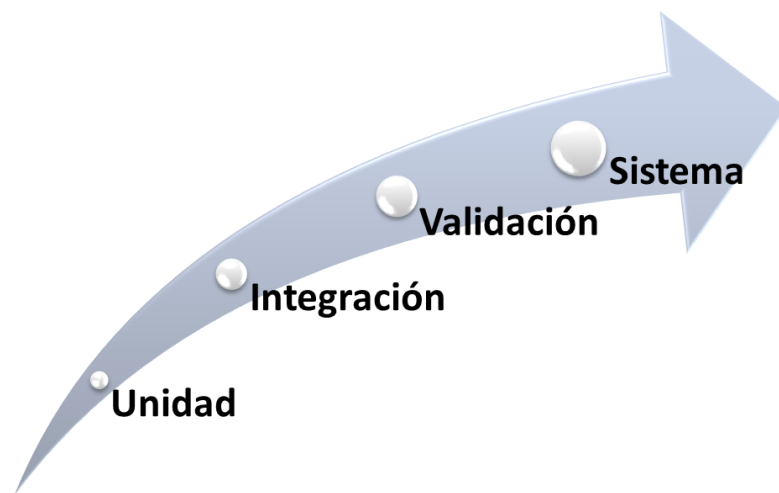


Fig. 15 Niveles de pruebas.

A medida que crece el nivel de las pruebas representadas, aumenta el alcance de las mismas; las pruebas de unidad, como su nombre lo indican, se concentran en cada unidad o componente del software; las siguientes son las pruebas de integración que verifican el diseño y la construcción de la arquitectura del software; a las que le siguen las pruebas de validación donde se validan los requisitos establecidos como parte del análisis de requisitos del software y por último las pruebas de sistema en las cuales se verifica el sistema como un todo. [23]

Las pruebas a desarrollar sobre el módulo Evaluación Criminológica son **las pruebas de unidad**, estas pruebas se concentran en la unidad más pequeña del software: el componente o el módulo y tienen como objetivo encontrar los errores dentro de los límites del módulo. [23]

Dentro de las pruebas de unidad se realizan pruebas de: interfaz, estructuras de datos locales, condiciones límites, rutas independientes y rutas de manejos de errores.

Se realizan las pruebas sobre la interfaz para verificar que la información fluye de la forma apropiada y en las direcciones necesarias en la unidad probada; también se prueban las estructuras de datos para verificar que los datos mantienen su integridad durante la ejecución de las funcionalidades; se recorren los caminos independientes o base para verificar que se hayan ejecutado al menos una vez; se prueban las condiciones límites para asegurar que el módulo funciona de la forma esperada ante los límites establecidos y por último se prueban los caminos para el manejo de errores. [23]

Luego de lo explicado anteriormente las pruebas son realizadas a las funcionalidades del módulo Evaluación Criminológica, este tipo de prueba examina si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ella se debe verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, validando los datos, ejecutando las pruebas y verificando el de comportamiento ante distintos escenarios, centrado dichas pruebas principalmente en la interfaz de usuario.

3.3.4. Diseño de casos de prueba para el CU Gestionar Información del Área Laboral.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1. Área Laboral 1.	EC 1.1 Área Laboral 1.	Recoge la pregunta 1 del Área Laboral realizada al interno. Si en algún momento de su vida ha trabajado.

	EC 1.2 Si en algún momento de su vida ha trabajado.	Cuando en la pregunta 1 se escoge la opción Si, el sistema muestra un formulario para recoger la edad de comienzo y los motivos para trabajar además de las preguntas 2, 3, 4 y 5 del Área Laboral realizadas al interno. Descripción de Empleos y Ocupaciones, con qué frecuencia llegaba tarde al trabajo, si alguna vez ha faltado al trabajo sin estar enfermo y si alguna vez ha estado desempleado.
	EC 1.3 Alguna vez ha faltado al trabajo sin estar enfermo.	Cuando en la pregunta 4 escoge la opción Si, el sistema muestra un formulario para recoger el motivo de la ausencia al trabajo sin estar enfermo.
	EC 1.4 Si alguna vez ha estado desempleado.	Cuando en la pregunta 5 se escoge la opción Si, el sistema muestra un formulario para recoger el motivo por el cual ha estado desempleado.
	EC 1.5 Faltan datos obligatorios.	Valida que se hayan llenado todos los campos obligatorios para avanzar hacia la siguiente página.
	EC 1.6 Datos incorrectos.	Valida que en el campo edad, se escriban sólo números.
SC 2. Área Laboral 2	EC 2.1 Área Laboral 2.	Recoge las preguntas 6 y 7 del Área Laboral realizadas al interno. De qué vivía cuando no estaba trabajando y si alguna vez alguien era responsable de su comida, dinero, casa, vestuario.

	EC 2.2 Alguna vez alguien era responsable de su comida, dinero, casa, vestuario.	Cuando en la pregunta 7 se escoge la opción Si, el sistema muestra un formulario para recoger quien era el responsable, cuándo, por cuánto tiempo, y el motivo.
	EC 2.3 Faltan datos obligatorios	Valida que se hayan llenado todos los campos obligatorios para avanzar hacia la siguiente página.
SC 3. Área Laboral 3	EC 3.1 Área Laboral 3.	Recoge las preguntas 8 y 9 del Área Laboral realizadas al interno. Si alguna vez se ha mantenido producto a resultados de actividades ilegales y otros aspectos de interés de esta área.
	EC 3.2 Alguna vez se ha mantenido producto al resultado de actividades ilegales.	Cuando en la pregunta 8, se escoge la opción Si, el sistema muestra un formulario para recoger por cuánto tiempo y qué actividades ilegales hacía.
	EC 3.3 Faltan datos obligatorios	Valida que se hayan llenado todos los campos obligatorios para avanzar hacia la siguiente página.

Tabla 2. Secciones a probar en el caso de uso.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	¿En algún momento de su vida ha trabajado?	Campo de selección única	No	Admite seleccionar sí o no.

[2]	Edad de comienzo	Campo de texto	No	Letras y números. En el formato (x años x meses) donde x es un número. Número de hasta 2 cifras.
[3]	Motivos para trabajar	Campo de texto	Si	Números y Letras. Hasta 100 caracteres.
[4]	Ocupación	Lista desplegable	No	Valores de la lista desplegable.
[5]	Lugar	Campo de texto	No	Números y Letras. Hasta 100 caracteres.
[6]	Actividad que realizó	Campo de texto	No	Números y Letras. Hasta 100 caracteres.
[7]	Tiempo	Campo de texto	No	Letras y números. En el formato (x años x meses) donde x es un número. Número de hasta 3 cifras.
[8]	Reconocimientos	Lista desplegable	No	Valores de la lista desplegable.
[9]	Sanciones	Lista desplegable	No	Valores de la lista desplegable.
[10]	Tiempo Desempleo	Campo de texto	Si	Letras y números. En el formato (x años x meses) donde x es un número. Número de hasta 3 cifras.

[11]	Motivo del cese	Campo de texto	Si	Números y Letras. Hasta 100 caracteres.
[12]	Frecuencia	Lista desplegable	No	Valores de la lista desplegable.
[13]	¿Por qué?	Área de texto	Si	Números y Letras. Hasta 255 caracteres.
[14]	¿Alguna vez ha faltado al trabajo sin estar enfermo?	Campo de selección única	No	Admite seleccionar sí o no.
[15]	¿Por qué?	Área de texto	Si	Números y Letras. Hasta 255 caracteres.
[16]	¿Alguna vez ha estado desempleado?	Campo de selección única	No	Admite seleccionar sí o no.
[17]	¿Por qué?	Área de texto	Si	Números y Letras. Hasta 255 caracteres.
[19]	¿De qué vivía cuando no estaba trabajando?	Área de texto	No	Números y Letras. Hasta 255 caracteres.
[20]	¿Alguna vez alguien era responsable de su comida, dinero, casa, vestuario?	Campo de selección única	No	Admite seleccionar sí o no.
[21]	¿Quién?	Campo de texto	No	Números y Letras. Hasta 100 caracteres.

[22]	¿A qué edad?	Campo de texto	No	Números y Letras. Hasta 100 caracteres.
[23]	¿Por cuánto tiempo?	Campo de texto	No	Letras y números. En el formato (x años x meses) donde x es un número. Número de hasta 3 cifras.
[24]	¿Por qué?	Área de texto	Si	Números y Letras. Hasta 255 caracteres.
[25]	¿Alguna vez se ha mantenido producto al resultado de actividades ilegales?	Campo de selección única	No	Admite seleccionar sí o no.
[25]	¿Por cuánto tiempo?	Campo de texto	No	Letras y números. En el formato (x años x meses) donde x es un número. Número de hasta 3 cifras.
[26]	¿Qué hacía?	Área de texto	No	Números y Letras. Hasta 255 caracteres.
[27]	Otros aspectos de interés de esta área	Área de texto	Si	Números y Letras. Hasta 255 caracteres.

Tabla 3 Descripción de las variables.

3.4. Resultados

En la realización de las pruebas participó el equipo de desarrollo, en las mismas se detectaron 14 no conformidades las que fueron resueltas por el equipo de desarrollo.

No	No conformidad	Ubicación	Estado
----	----------------	-----------	--------

1	No se guarda la entrevista.	evaluacioncriminologica/ entrevistas/ criminológica/ personal/ areaCognitiva5.gsp	Resuelta
2	Los mensajes de error no aparecen en la página correcta.	evaluacioncriminologica/ entrevistas/ criminológica/ criminológica/ comportamientoReclusion.gsp	Resuelta
3	Los mensajes de error no aparecen en la página correcta.	evaluacioncriminologica/ entrevistas/ criminológica/ criminológica/ comportamientoReclusion1.gsp	Resuelta
4	Botón Atrás no funciona.	evaluacioncriminologica/ entrevistas/ criminológica/ social/ areaEducativa2.gsp	Resuelta
5	Cuando se presiona el botón Atrás, el campo “¿Qué tiempo demora en calmarse?” no carga el valor que se le asignó anteriormente.	evaluacioncriminologica/ entrevistas/ criminológica/ personal/ areaCognitiva3.gsp	Resuelta
6	Cuando se presiona el botón Atrás, el campo “¿Quién inicia la riña?” no carga el valor que se le asignó anteriormente.	evaluacioncriminologica/ entrevistas/ criminológica/ personal/ areaCognitiva4.gsp	Resuelta
7	El campo de selección	evaluacioncriminologica/	Resuelta

	“Empleos” admite que no se seleccione nada.	entrevistas/ criminológica/ social/ areaLaboral1.gsp	
8	El campo de selección “Reconocimientos” admite que no se seleccione nada.	evaluacioncriminologica/ entrevistas/ criminológica/ social/ areaLaboral1.gsp	Resuelta
9	El campo de selección “Sanciones” admite que no se seleccione nada.	evaluacioncriminologica/ entrevistas/ criminológica/ social/ areaLaboral1.gsp	Resuelta
10	Campo “Otros aspectos de interés” validado como obligatorio.	evaluacioncriminologica/ entrevistas/ criminológica/ social/ areaLaboral3.gsp	Resuelta
11	Campo “Otros aspectos de interés” validado como obligatorio.	evaluacioncriminologica/ entrevistas/ criminologica/ social/ areaFamiliar5.gsp	Resuelta
12	Campo “Otros aspectos de interés” validado como obligatorio.	evaluacioncriminologica/ entrevistas/ criminologica/ social/ antecedentes3.gsp	Resuelta
13	Campo “Otros aspectos de interés” validado como obligatorio.	evaluacioncriminologica/ entrevistas/ criminologica/ social/ areaEducativa3.gsp	Resuelta
14	Campo “Otros aspectos de interés” validado como obligatorio.	evaluacioncriminologica/ entrevistas/ criminologica/ criminologica/ habitosToxicos3.gsp	Resuelta

Tabla 4 Resultados de las Pruebas.

3.5. Conclusiones Parciales.

En este capítulo se mostraron y explicaron los artefactos generados durante la fase de implementación, el diagrama de despliegue, el diagrama de componentes y la implementación del sistema. También se explicaron las características de la creación de entidades y algunas de las funciones que facilitan el trabajo con el modelo de datos. Además se realizaron las pruebas necesarias para demostrar que el sistema funciona correctamente y que cumple con los requisitos especificados. Se detectaron 14 no conformidades las cuales fueron resueltas por el equipo de desarrollo del módulo.

Conclusiones

Como resultado de la realización de este trabajo, se diseñó e implementó el módulo Evaluación Criminológica del SIDE P dotando así al sistema de un mecanismo que facilita la gestión del proceso de evaluación criminológica en los Centros Penitenciarios. Además se le ha dado cumplimiento a los objetivos propuestos inicialmente obteniendo los siguientes resultados:

- Se realizó un análisis de soluciones informáticas existentes para los Sistemas Penitenciarios tanto nacional como internacional.
- Se describieron las herramientas y tecnologías a utilizar para la implementación del módulo.
- Se generaron los artefactos relacionados con el diseño y la implementación del módulo Evaluación Criminológica.
- Se realizaron las pruebas correspondientes al módulo obteniendo un total de 14 no conformidades las cuales fueron resueltas en su totalidad.

Recomendaciones

Se recomienda la utilización de un Sistema Experto que, luego de haber recopilado la información mediante las entrevistas, contribuya con el proceso de toma de decisiones para la asignación de pronósticos de evolución.

Referencias Bibliográficas

1. manual de dojo. [En línea] <http://www.dojotoolkit.org>.
2. manual de grails. [En línea] <http://www.grails.org>.
3. **Arias Manuel, Álvarez José R.** Universidad Nacional de Educación a Distancia, España. [En línea] <http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node60.html>.
4. *Breve esbozo de la criminología en Cuba.* **Castro, Dra. Lolita Aniyar de.** 1995. Maracaibo, Venezuela. : Edición Especial 23.2.95 de Capítulo Criminológico, 1995.
5. **Brito, Nacho.** 2009. Manual de desarrollo web con Grails. [En línea] junio de 2009. <http://www.manual-de-grails.es> .
6. **Christopher M. Judd, Joseph Faisal Nusairat, James Shingler.** 2008. *Biginning Groovy and Grails From Novice To Professional.* New York : s.n., 2008.
7. Concepto de JavaScript. [En línea] <https://developer.mozilla.org>.
8. *Criterios de selección de metodologías de desarrollo de software.* **Tinoco Gómez, Ing. Oscar, Rosales López, Ing. Pedro Pablo y Salas Bacalla, Ing. Julio.** 2010. 13, Perú : UNMSM, Revista de la Facultad de Ingeniería Industrial, 2010, Vols. 70-74. ISSN: 1810-9993.
9. **Cupertino, J.R.** 1998. *El Lenguaje Unificado de Modelado.* California : s.n., 1998.
10. Embarcadero Technologies. [En línea] <http://www.embarcadero.com>.
11. 2012. Entorno Virtual de Aprendizaje- Curso Programación IV. [En línea] 2012. eva.uci.cu.
12. 2009. EPVNET: Project Web Hosting - Open Source Software. *EPVNET: Project Web Hosting - Open Source Software.* [En línea] 2009. <http://epvnet.sourceforge.net/>.
13. 2012. EVA. [En línea] 2012. eva.uci.cu/mod/resource/view.php?id=9066.
14. **Glen Smith, Peter Ledbrook.** 2009. *Grails in Action.* Greenwich : Manning Publications Co., 2009.
15. *Glosario de nuevas definiciones en el Reglamento del Sistema Penitenciario.*
16. Groovy. [En línea] <http://groovy.org.es/>.
17. **Hernández Aguilar, Ing. Violena y González Jorrín, Ing. Michael.** Entorno Virtual Aprendizaje UCI. [En línea] eva.uci.cu/file.php/160/Curso_2010-2011/Semana_10/Clase_Practica_6/Materiales_Basicos/Material_EI_Como_disenar_casos_de_pruebas_a_partir_de_los_casos_de_uso.pdf.
18. *Informe de factibilidad del proyecto de desarrollo estratégico del Sistema Penitenciario Nacional.*
19. *Informe Factibilidad Modernización SPN (versión 2).*
20. JavaScript. [En línea] http://www.java.com/es/download/faq/java_javascript.xml.

21. **Kroll, Per y Kruchten, Philippe. 2007.** *The Rational Unified Process made Easy*. United State : Pearson Education, Inc., 2007. ISBN: 0-321-16609-4.
22. **LLC, LockWorks. 2007.** Jail and Prision Management Software. *Lock&Track Corrections Information System*. [En línea] 09 de mayo de 2007. <http://www.locktrack.com/>.
23. **Pressman, Roger S. 2005.** *Ingeniería del Software, Un enfoque práctico*. s.l. : McGraw-Hill, 2005. ISBN: 9701054733.
24. *Reglamento del Sistema Penitenciario*.
25. **S.A, Albet. 2010.** *DESCRIPCIÓN DEL PRODUCTO SIGEP V 2.1*. Ciudad de La Habana : s.n., 2010. PR4-SW-DE-059.
26. **Seth Ladd, Darren Davison, Steven Devijver, Colin Yates and Keith Donald. 2006.** *Expert Spring MVC and Web Flows*. United States of America : Springer-Verlag, 2006. ISBN-13 (pbk): 978-1-59059-584-8.
27. Sitio de Grails. [En línea] <http://grails.org/Documentation>.
28. Sitio Oficial de Java. [En línea] <http://www.java.com/>.
29. Sitio Oficial de Oracle. [En línea] <http://www.oracle.com>.
30. Sitio Oficial NetBeans. [En línea] www.netbeans.org.
31. **2010.** SYSCON Justice Systems. *SYSCON Justice Systems*. [En línea] Syscon, 2010. <http://www.syscon.net/Products/elite-jails>.
32. **Taylor, E. S. 1959.** *An Interim Report on Engineering Design*. Massachusetts : Massachusetts Institute of Technology, 1959.
33. The Apache Software Foundation. [En línea] <http://tomcat.apache.org>.
34. TortoiseSVN the coolest interface to (Sub)version control. [En línea] tortoisesvn.net/docs/release/TortoiseSVN_es/tsvn-intro-features.html.
35. **2011.** Visual Paradigm: Boost Productivity with Innovative and Intuitive Technolgies. *Visual Paradigm: Boost Productivity with Innovative and Intuitive Technolgies*. [En línea] VP Teamwork, 2011. <http://www.visual-paradigm.com/product/vpuml/provides/>.

Anexos

Anexo 1: Modelo de Datos

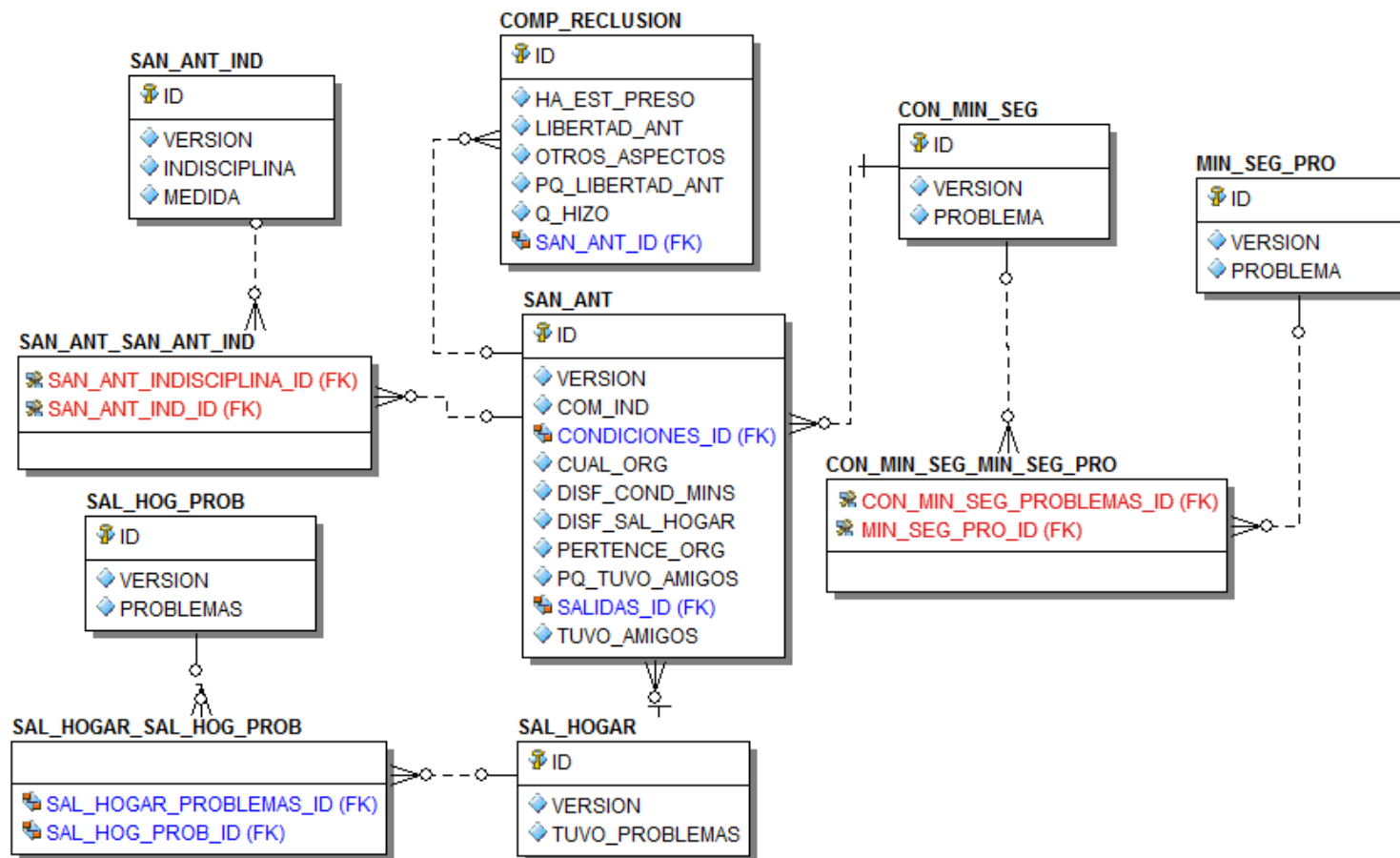


Diagrama 1. Entidades pertenecientes al CU Gestionar Información de Antecedentes de Padecimiento.

Anexo 2: Diagramas del diseño

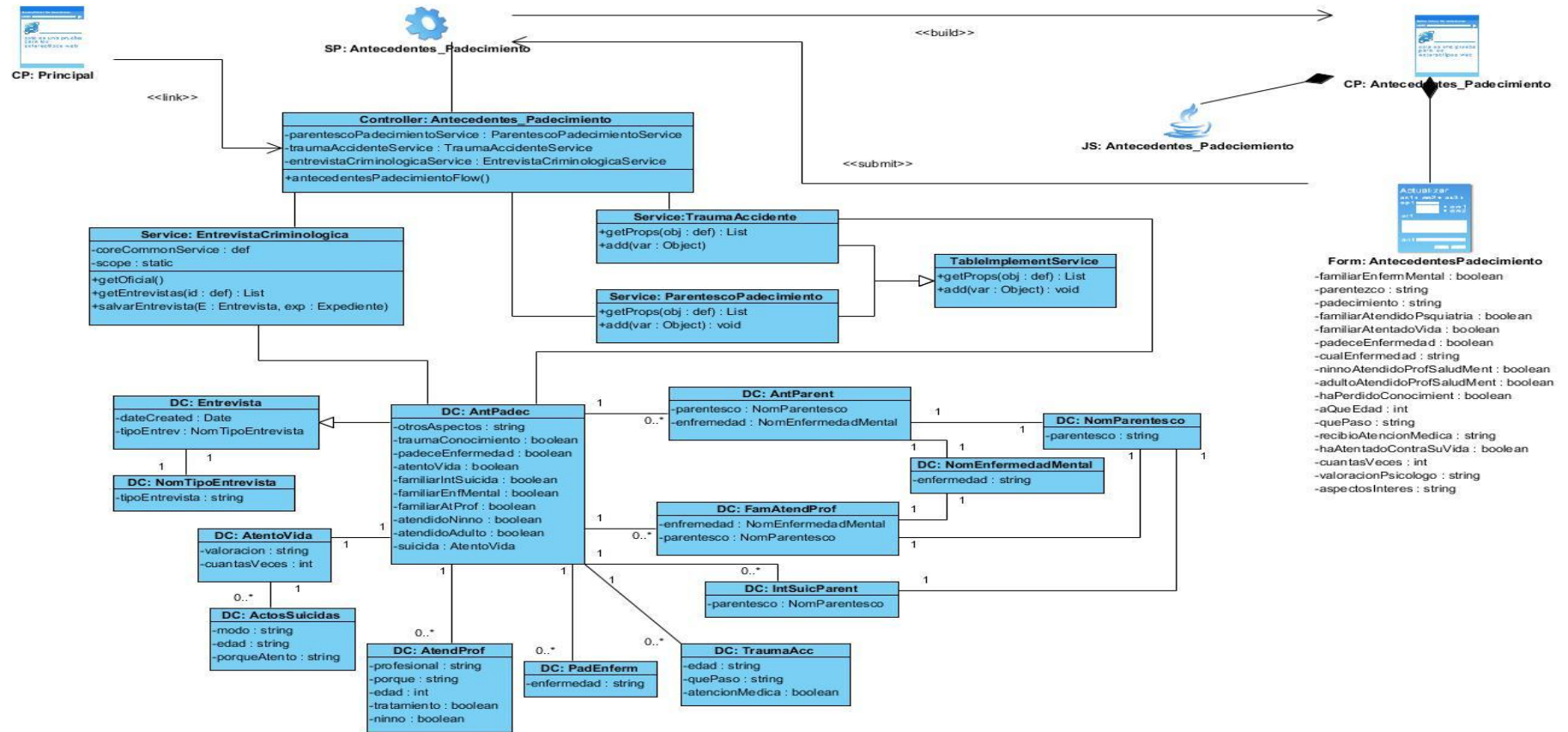


Diagrama 2. Diagrama de Clases del Diseño (DCD) Antecedentes de Padecimiento.

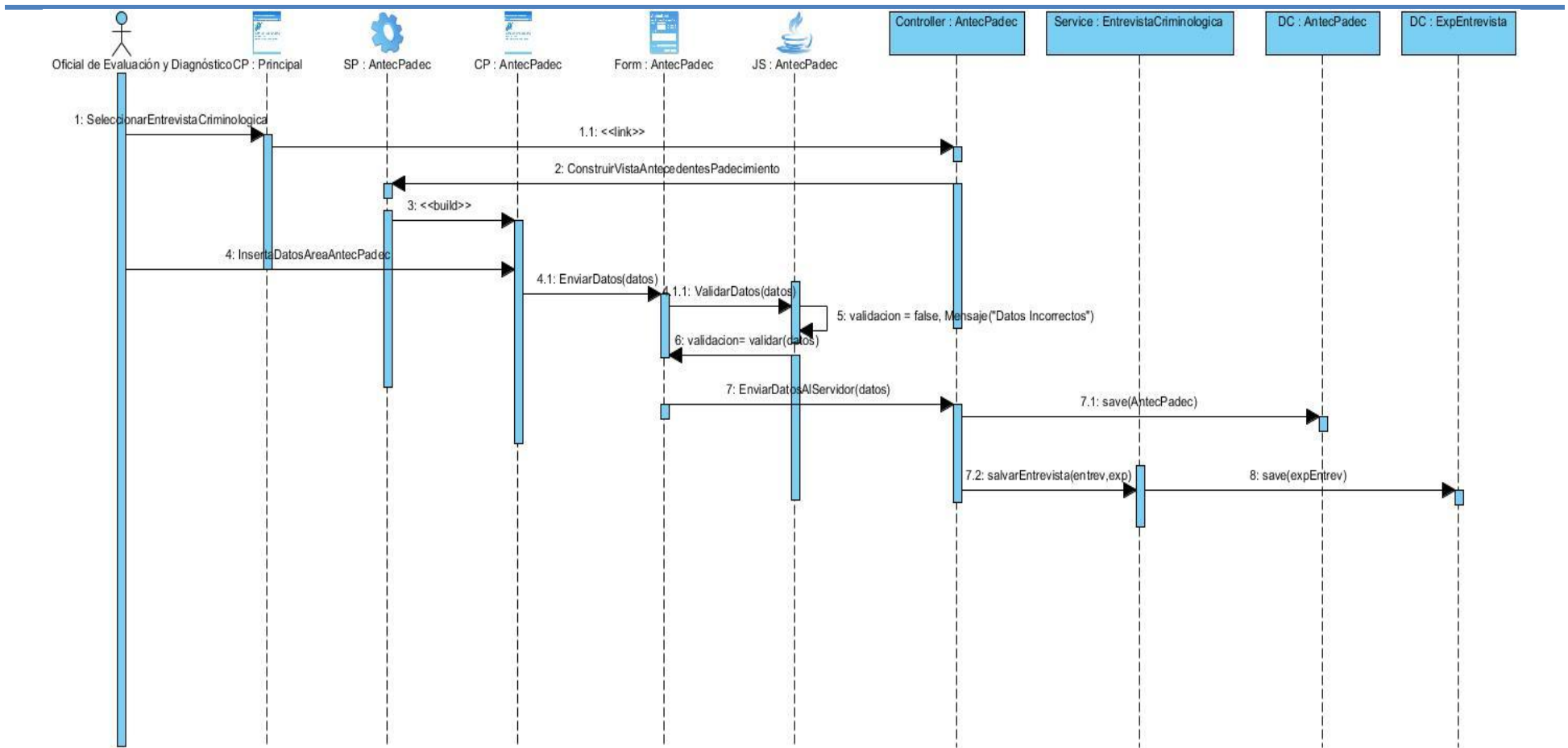


Diagrama 3. Diagrama de Secuencia (DS) Registrar Antecedentes de Padecimiento.

Glosario de Términos

AJAX¹²

AJAX es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Framework Spring

Spring es un framework de Software Libre para Java. Su principal característica es brindar una fábrica de objetos utilizando la inyección de dependencias.

DAO

Data Access Object define la relación entre la lógica de presentación de una aplicación y los datos.

GoF

Gang of Four hace referencia al nombre con el que comúnmente se conoce a los autores del libro *Design Patterns*. Estos patrones de diseño se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

GRASP

Son Patrones Generales de Software para Asignación de Responsabilidades (General Responsibility Assignment Software Patterns), aunque son considerados que más que patrones propiamente dichos pues constituyen una serie de buenas prácticas recomendables en el diseño de software.

Framework Hibernate

Es una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.

¹² Asynchronous JavaScript And XML, JavaScript asíncrono y XML

Criteria

Permite la creación de consultas personalizadas a la base de datos basadas en un Builder de Groovy y la Criteria API de Hibernate.